Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Usando o Twitter para detecção de usuários vulneráveis a phishing

Jefferson Viana Fonseca Abreu

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Jorge Henrique Cabral Fernandes

Brasília
2018

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Usando o Twitter para detecção de usuários vulneráveis a phishing

Jefferson Viana Fonseca Abreu

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Jorge Henrique Cabral Fernandes (Orientador)
CIC/UnB

Prof. Dr. João José Costa Gondim
CIC/UnB

Me. Raul Carvalho de Souza
Procuradoria Geral do Distrito Federal

Prof. Dr. Edison Ishikawa
Coordenador do Bacharelado em Ciência da Computação

Brasília, 24 de dezembro de 2018

Dedicatória

Ao leitor.

Agradecimentos

Agradeço a todos que de alguma maneira colaboraram com a elaboração deste trabalho. Em especial para minha mãe Geni que desde cedo me ensinou a valorizar a incansável busca pelo conhecimento, ao meu orientador o Dr. Jorge Fernandes que sempre que possível esteve disponível para compartilhar o seu conhecimento, a minha estimada Judi que esteve sempre ao meu lado me incentivando quando necessário e ao presidente Lula que democratizou o acesso ao ensino superior público e de qualidade me permitindo sonhar mais alto.

Resumo

Este trabalho relata pesquisa sobre o desenvolvimento de um modelo matemático-computacional para detecção de usuários de contas do *twitter* que podem ser vulneráveis a *phishing*. Por meio da condução de uma pesquisa exploratória e descritiva, de abordagem quantitativa, embasada por uma série de experimentos, verificou-se a existência de possíveis correlações entre a autoridade (estruturalmente calculada em um grafo) de uma conta no *twitter*, entre outros atributos, e a vulnerabilidade de seu usuário a um ataque de *phishing*. Foram desenvolvidas 4 versões de um experimento de coleta de dados, envolvendo a realização de pseudo-ataques em aproximadamente 1287 contas, ao longo de 38 dias. Os resultados foram analisados por meio de regressão logística. A análise concluiu que é possível realizar uma perfilização com os atributos que levam supostas vítimas a caírem no golpe, possibilitando ataques com uma precisão melhor que a escolha aleatória, além disso é possível construir uma ferramenta que realiza ataques de engenharia social automatizada no *twitter*.

Palavras-chave: Phishing, engenharia social, BOTS, vulnerabilidades, twitter

Abstract

This works reports the development of a mathematical-computational model to perform the detection of twitter accounts who may be vulnerable to phishing attacks. Through descriptive, exploratory and quantitative research based on a series of experiments, it verifies the existence of correlations between some attributes of twitter accounts and vulnerability of their users to phishing attacks. Four different incremental versions of experiments were performed, envolving fake phishing attacks directed to approximately 1287 twitter accounts extended across 38 days. The results were analysed through logistic regression. Analysis, confirm that it's possible perform a victm profiling based on account attributes, making possible do attacks with precision better than a random choice, and moreover, that is possible to build and run a tool to perform automated social engineering attacks.

Keywords: Phishing, social engineering, BOTS, vulnerability, twitter

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

O uso de computadores conectados à Internet tem produzido uma série de avanços na forma como a humanidade se estrutura e funciona, mas essas transformações não ocorrem sem a exposição de seus usuários a novos riscos que não existiam em um mundo previamente analógico. O controle desses riscos tem sido feito por meio da implantação de controles e políticas de segurança em ambientes computacionais, sendo muitos desses controles feitos por artefatos computacionais, enquanto outros são promovidos com a transformação comportamental dos usuários. Como atuar para gerenciar vários desses riscos, especialmente no que concerne aos riscos e ataques de engenharia social? A pesquisa aqui relatada propõe caminhos para uma melhor utilização de um subconjunto dos dados produzidos por um tipo de serviço bastante popularizado no ambiente de computadores conectados à Internet, que é o *Twitter*. O objetivo é promover melhor compreensão do comportamento inseguro de usuários no evento de ataques de engenharia social.

Antes de apresentar ao final, e de forma mais precisa, a motivação para o trabalho, esta introdução apresenta brevemente os principais conjuntos de elementos que abrem o contexto para a realização do trabalho:

- A dificuldade para desenvolvimento de políticas de segurança da informação;
- O surgimento do *Phishing* enquanto técnica de engenharia social; e
- Twitter como fonte de dados abertos sobre comportamentos do usuário.

1.1 Computadores, Internet e políticas de segurança da informação

Os computadores e a Internet permitiram que parte da humanidade executasse de maneira mais automática várias tarefas repetitivas, economizando recursos humanos (RH)

que podem ser valiosos para organizações públicas e privadas. Um exemplo no contexto brasileiro é o caso dos bancários, onde os bancos conseguiram economizar RH com o advento dos terminais de autoatendimento. É evidente também que os computadores guardam cada vez mais informações sensíveis, não só de grandes corporações ou governos, mas também de usuários domésticos [1, p.379]. Alguns exemplos de informações sensíveis de usuários domésticos são registros de operações financeiras, fotos, registros de conversas etc. Portanto, se por um lado cada vez mais os computadores e a Internet permitem a automação de atividades vitais, por outro eles armazenam informações valiosas que podem vir a ser alvo de pessoas malintencionadas, e por isso necessitam de políticas de segurança da informação concernentes ao seu uso.

De acordo com [2, p.8] na norma ABNT NBR ISO/IEC 17799:2005 políticas de segurança da informação são diretrizes criadas com o objetivo de "Prover uma orientação e apoio da direção para a segurança da informação de acordo com os requisitos do negócio e com as leis e regulamentações relevantes."

Apesar da adoção dessas políticas de segurança da informação vislumbrarem um ambiente computacional substancialmente seguro [2] , não podemos desprezar o fator humano, por muitos considerado o elo mais fraco do processo da segurança [3, p. 3] [4, 5, 6, 7], e que por tantas vezes é a prevalente causa da insegurança [4]. Por exemplo, não é de muita valia uma organização investir tempo e dinheiro na implementação de fortes políticas e controles técnicos de segurança da informação, se seus colaboradores estão abertos a fornecer suas credenciais de acesso para terceiros para os quais não se pode ter certeza absoluta da benevolência de suas intenções. Um dos mecanismos que pode ser usado para o ganho de vantagens indevidas sobre seres humanos é a engenharia social. Trata-se de um processo social/psicológico, onde uma pessoa (atacante) obtém informação de um alvo em específico através de uma outra pessoa [8]. Isso é possível através do uso de influência, persuasão, personificação, ou por pura manipulação [3, p. iv].

1.2 Engenharia social

A engenharia social consiste em alguém (doravante chamado atacante) usar a influência e/ou persuasão para ludibriar pessoas e convencê-las de que o atacante é outra pessoa, ou manipular as pessoas para tomar decisões as quais essas não tomariam normalmente [3]. Por exemplo, fornecer acesso em determinado sistema, conceder a senha pessoal de autenticação, fornecer informações pessoais de colaboradores de uma organização, etc. Através da engenharia social é possível se aproveitar das pessoas com (em golpes onde há o contato pessoal com a vítima) ou sem (por exemplo, através de *phishing*) o uso da tecnologia digital.

Existem diversos meios pelos quais a engenharia social pode ser exercida, e na contemporaneidade, além do contato direto com a vítima, os atacantes podem recorrer a vários outros artifícios como o uso *spoofing* (falsificação de identidade digital) para aumentar a credibilidade na personificação de uma ligação telefônica [9], mensagem de *e-mail* [10, 11], ou um site falso [12]. Há também o uso de ferramentas de engenharia social automatizada [13, 14, 7, 11], ou a simples obtenção de informações vasculhando lixo [3], espreitando por cima dos ombros de um usuário legítimo enquanto este utiliza um computador [3, 15], com o uso de *phishing* [9] e muitas outras técnicas [9, 3].

1.3 *Phishing*

Dentre o leque de técnicas de engenharia social citadas, este trabalho aborda o *phishing*. Este é um tipo de golpe digital muito presente no cotidiano dos usuários da rede mundial de computadores. Entretanto, apesar de ser uma fraude muito comum, não há um consenso sobre sua definição, esta falta de consenso pode causar a confusão nas pessoas ao lidar com o assunto, além de dificultar o desenvolvimento de pesquisa na área [16].

Em 2014, Lastdrager [16] notou essa lacuna de conhecimento e conduziu uma revisão sistemática de literatura para procurar uma definição para *phishing* que fosse comum e aplicável a todos os usos do termo, e por isso, o pesquisador juntou todas essas definições em uma única definição que fosse comum à maioria. Destarte as pesquisas envolvendo o *phishing* poderiam utilizar esta definição e assim estar mais alinhadas na descrição do problema.

De acordo com [17], o *phishing* pode ser definido como enganar usuários para que esses revelem informações sensíveis através de meios baseados em computador. Já Lastdrager definiu o *phishing* como uma fraude escalável onde a personificação é usada para obter informações de um alvo [16]. Para isso, o atacante precisa conduzir um usuário para um *website* de coleta de dados fraudulento [18, 17], onde este pode vir a ser vítima de um golpe. No *phishing*, o atacante fornece uma isca, objetivando que alguém caia na armadilha e permitindo assim que quem está executando o golpe obtenha algum tipo de vantagem indevida [15], com o uso dos dados obtidos. Existem diversos meios nos quais a isca pode ser disseminada para alvos, um destes pode ser através das mídias sociais *online*.

1.4 *Twitter*

No presente momento, o uso de mídias sociais *online* (doravante denominadas apenas por mídias sociais) é bem comum, e é notável que uma parcela da população mundial faz uso

cotidiano dessas mídias. Dentre as diversas opções de mídias sociais, podemos destacar o Twitter, bastante popular em todo o mundo, o que pode ser demonstrado pelas 319 milhões de contas de usuários ativas por mês no ano de 2016, o Brasil foi um dos países que apresentou maior crescimento de contas no *Twitter* [19].

A relevância desta mídia social é tão grande, que diversas celebridades, pessoas notáveis e chefes de estado adotam esta plataforma como meio de comunicação principal com suas respectivas audiências. Seria ela útil para detectar comportamentos vulneráveis? Dois trabalhos evidenciam a utilidade do uso desses dados.

Em um primeiro, os pesquisadores [13] desenvolveram um trabalho para mostrar como o uso de *Social Bots* - robôs programados em software - que tentam simular o comportamento humano de envio de postagens no Twitter pode representar riscos para organizações. No experimento, foram construídos 8 *Social Bots* que faziam postagens no Twitter com temas selecionados pelos pesquisadores, visando conduzir usuários do twitter para web sites de *phishing* através de um *link* malicioso. Após 2 semanas de experimentação, a equipe de pesquisadores obteve entre 410 e 417 acessos ao *website* de *phishing* da pesquisa, onde dentre esses pelo menos 33 eram provenientes de usuários de redes de empresas, mostrando que o uso de *Social Bots* é uma ameaça real à segurança computacional.

O trabalho de Shafahi [13] evidencia que os dados gerados pelo twitter podem ser importante fonte de informação para a identificação de usuários vulneráveis a ataques de engenharia social, seja para a melhora das políticas de segurança, seja para finalidade de realização de ataques. Um dos pressupostos deste trabalho é referente ao primeiro tipo de aplicação.

Em outro trabalho, os pesquisadores Nagmoti et al. [20] definiram um método computacional para mensurar a autoridade de uma conta no twitter, isso é, uma conta que produz tuítes. Baseando-se nas medidas de *TweetRank* e *FollowerRank*, eles conseguiram estimar a relevância de um tuíte em particular. As fórmulas foram validadas por usuários humanos, verificando se os tuítes considerados relevantes pelas pessoas coincidiam com os indicados pelo método desenvolvido. O experimento constatou que o método é aplicável ao ranqueamento um fluxo de tuítes (*stream*) em tempo real, isso é, à medida em que os tuítes são enviados pelos usuários.

Além disto a medida do *FollowerRank* pode ser interpretada como a autoridade que uma conta possui, uma vez que contas com essa medida alta tendem a produzir conteúdo mais relevante para usuários humanos [20]. A hipótese dos autores desta pesquisa é que um dos fatores que pode influenciar a maior ou menor efetividade de um ataque de *phishing* no Twitter, provocando pelo envio de tuítes, é o *FollowerRank*, aqui interpretado como autoridade.

1.5 Motivação

Já existem demonstrações claras de que as atuais ferramentas anti-*phishing* não funcionam de maneira eficiente [12, 15]. Entretanto, alguns estudos já demonstraram que o trabalhos que objetivam educar as pessoas contra essa ameaça apresentam bons resultados [21, 15]. Assim sendo, o pressuposto deste trabalho é que, para se obter resultados mais impactantes na luta contra o *phishing*, é necessário identificar quais são os grupos de usuários mais vulneráveis, para saber onde medidas educacionais merecem maior foco. Dado que há uma limitada quantidade de recursos para serem aplicados em ações educacionais, onde aplicar os recursos da melhor forma? Onde investir para maximizar retorno? Possivelmente junto aos usuários que apresentam maior vulnerabilidade.

Analisando os fatos supracitados, os pesquisadores conceberam algumas perguntas de pesquisa, às quais este trabalho objetiva responder, através de uma pesquisa descritiva e quantitativa:

- p1: É possível identificar alguns grupos de contas mais vulneráveis a ataques de *phishing* dentro do Twitter?
- p2: A autoridade de uma conta(tomada como *FollowerRank*) no twitter, ou os demais atributos de uma conta no twitter, possuem correlação com a propensão de seu usuário a fornecer informações a desconhecidos? O assunto de interesse da conta afeta essa correlação?

1.6 Objetivo geral e objetivos específicos

Esta pesquisa foi elaborada para fins meramente didáticos, vislumbrando testar conceitos sobre a engenharia social. Todos os ataques aqui executados durante os experimentos foram feitos da maneira mais inócua possível. O objetivo geral foi desenvolver um modelo matemático-computacional para explicar o fenômeno da vulnerabilidade do usuário de uma conta de ser indiscreto no fornecimento de dados sensíveis na sua relação com estranhos, na mídia social Twitter. Para fornecer respostas às questões, o trabalho conduz pseudo-ataques de *phishing*, junto a contas de usuários do Twitter, cuja autoridade pode ser calculada previamente, a fim de verificar se há uma correlação entre a autoridade dessa conta e a propensão de seu usuário em fornecer informações supostamente sensíveis a um desconhecido (o pseudo-atacante). Outras correlações também são exploradas. Para possibilitar essa observação, o trabalho é subdividido em objetivos específicos, e mediante o cumprimento destes, vislumbra-se alcançar o objetivo principal da pesquisa:

1. Investigar quantitativamente (através de métodos estatísticos) se existe uma correlação entre a autoridade de uma conta no twitter, e outros atributos, e a propensão do usuário dessa conta a fornecer informações supostamente sensíveis a desconhecidos;

2. Investigar quantitativamente (através de métodos estatísticos) se existe uma correlação entre o tipo de assunto discutido no twitter a propensão do usuário dessa conta a fornecer informações supostamente sensíveis a desconhecidos;

1.7 Justificativa

Vislumbrando a criação de ambientes computacionais substancialmente mais seguros, é relevante que sejam neutralizados mecanismos que objetivam fornecer vantagens indevidas para terceiros. Porém, de nada adianta possuir um computador inexpugnável, munido com os mais avançados controles de segurança computacional, se os usuários humanos estão suscetíveis a outros tipos de trapaças, como serem ludibriados a compartilhar informações sensíveis com estranhos. Uma dessas ameaças é o *phishing*, e é um problema já conhecido a dificuldade de detectar este tipo de investida [15, 12, 9].

Para contornar essa adversidade no mundo virtual, especialmente nas mídias sociais, é necessário o desenvolvimento de técnicas que visam combater ataques desse tipo. Porém, para elaborar esses métodos de maneira eficaz, necessitamos primeiramente conhecer a ameaça, e infelizmente como já foi supracitado neste trabalho, há uma carência de pesquisas sobre a efetividade de vetores - canais ou meios que são usados para conduzir ataques, como os tuítes - diferentes para o *phishing* [13]. Tais pesquisas possibilitariam melhor quantificar o número de seres humanos em risco de serem ludibriados, e o que faz com que isso aconteça [12]. Para realizar essa tarefa, pesquisas científicas são de suma importância, pois dessa forma é possível realizar experimentos em ambiente controlado, com menores riscos aos alvos.

As ferramentas *anti-phishing* atuais não funcionam de maneira eficiente, em um exercício de pseudo-ataques utilizando *e-mail*, cerca de 10% a 11% dos ataques chegariam aos alvos, mesmo passando pelos filtros presentes nos servidores de *e-mail* [12]. Segundo [15] vários trabalhos já mostraram que nem soluções computacionais (controles de segurança técnicos) no lado do aplicativo cliente (ex: navegador web), nem soluções computacionais no lado do servidor (ex: sítio web), são bem sucedidas no que se trata de evitar que um usuário vulnerável seja enganado.

Entretanto, alguns estudos já demonstraram que o trabalhos que objetivam educar as pessoas contra a ameaça do *phishing* apresentam bons resultados. Em um estudo com empresas do Recife, o treinamento intensivo mostrou resultados positivos no que se trata da prevenção do *phishing* [21]. Existem vários outros trabalhos mostrando que medidas

educativas são efetivas contra o *phishing* [15]. Por isso, vislumbrando obter resultados mais impactantes na luta contra o *phishing*, é necessário identificar quais são os grupos mais vulneráveis para saber onde as medidas educacionais merecem maior foco.

Com base nas questões, pressupostos e hipóteses apresentadas neste capítulo, a pesquisa aqui relatada estruturou-se na forma dos cinco capítulos restantes, a seguir apresentados. O capítulo 2 apresenta uma revisão teórico-conceitual sobre (1) o desenvolvimento e uso de ferramentas de coleta de dados para a mídia social *twitter*, (2) o porquê da autoridade ser um bom indicador de propensão a compartilhamento de dados com estranhos, (3) técnicas para a construção de ataques de engenharia social e *phishing*, e (4) a importância da sensibilização de usuários para os ataques de engenharia social. O capítulo 3 apresenta o percurso metodológico seguido pelos pesquisadores, desde as primeiras ideias desenvolvidas, até a realização dos experimentos, problemas encontrados e alternativas construídas. O capítulo 4 faz uma apresentação detalhada dos resultados obtidos com os vários experimentos realizados. O capítulo 5 analisa e discute os resultados obtidos com base no referencial teórico conceitual apresentado no capítulo 2. O capítulo 6 verifica o alcance dos objetivos traçados, e quais os problemas encontrados e soluções propostas, bem como traça perspectivas de trabalhos futuros.

Capítulo 2

Revisão Teórico-Conceitual

Este capítulo define conceitos que são fundamentais para o total entendimento do trabalho científico desenvolvido. Primeiro são abordados alguns tópicos aprofundando o que foi apresentado na introdução. Esses tópicos versam sobre politicas de segurança da informação, engenharia social e *phishing*. Depois, baseando-se no conteúdo apresentado é discutido o que são sistemas de engenharia social automatizada. Em seguida é aprofundado o que é a mídia social *twitter* e como funciona o desenvolvimento de aplicativos para a plataforma.

2.1 Políticas de segurança da informação

Os sistemas de computadores, interconectados por meio da Internet e seus protocolos, já são um elemento fundamental para a gerência (e armazenamento) de informações significativas e/ou sigilosas, seja para as organizações, seja para os indivíduos. É de conhecimento geral que cada vez mais o mundo está mais dependente de computadores [22]. Por exemplo, cada vez mais temos substituído registros e documentos em papel por registros e documentos digitais. No Brasil atualmente é possível utilizar a carteira de habilitação, cadastro de pessoa física e título de eleitor virtuais [23]. É comum utilizar dispositivos computacionais interligados a internet para fazer compras ou realizar serviços bancários. Na atualidade inclusive, existem empreendimentos que não possuem escritório físico e estão baseados em grande parte na internet, por exemplo, estabelecimentos de *e-commerce*, blogs, cursos online e etc.

No ponto de vista da segurança da informação, os sistemas computacionais possuem três propriedades gerais: confidencialidade dos dados, integridade dos dados e disponibilidade do sistema. Entretanto, alguns computadores ultimamente estão sendo invadidos para que sejam transformados em zumbis que servem como escravos para um invasor [1, p.380], e existem ameaças que vislumbram impedir que os sistemas computacionais cum-

Objetivo	**Exemplo de ameaças**
Confidencialidade dos dados	Exposição dos dados para pessoas não autorizadas.
Integridade dos Dados	Alteração indesejada dos dados.
Disponibilidade do sistema	Negação de serviços.
Exclusão de invasores	Controle do sistema por pessoas não autorizadas.

Tabela 2.1: Principais objetivos de segurança e suas principais ameaças. Fonte: Adaptado de [1, p. 380]

pram com seus objetivos, como é exemplificado na tabela 2.1, que contrapõe os objetivos da segurança da informação com as respectivas ameaças ao cumprimento dos objetivos.

Entretanto, para cada ameaça citada na tabela 2.1 existem políticas de segurança que podem atenuar ou até mesmo eliminá-las. Por exemplo o uso de bons protocolos criptográficos pode até mesmo eliminar totalmente, com o custo computacional inerente ao processo, a exposição dos dados para pessoas não autorizadas. A adoção de fortes políticas de segurança de redes pode dificultar severamente a intrusão de pessoas indesejadas em algum sistema computacional. Por isso é necessário que cada usuário e gestor de ambiente computacional saiba mensurar a importância que o computador representa em sua vida, para que destarte possa saber quais políticas de segurança são necessárias e viáveis na sua interação com a máquina.

2.2 Um pouco mais sobre engenharia social

A engenharia social não é uma ameaça nova, nem tampouco é algo exclusivo do ambiente computacional. O uso desse termo primeiramente foi feito no ramo da ciência política e não necessariamente estava ligado a algo nocivo [9]. Inicialmente, os engenheiros sociais eram "engenheiros sociais e políticos". Eram políticos, economistas, e sociólogos, pessoas capazes de moldar como uma sociedade se comportaria, através de 3 mecanismos [9].

1. Assimetria epistêmica: A diferença de conhecimentos entre o engenheiro social e a sociedade;
2. Dominância tecnocrática: O domínio de conhecimento técnico por parte do engenheiro social; e
3. Substituição teleológica: a capacidade que o engenheiro social possui modificar os objetivos de uma sociedade conforme desejar.

O termo Engenharia Social começou a ser vinculado à segurança cibernética com o advento da subcultura dos *phone phreakers* [3]. Esta consistia em grupos de pessoas que exploravam o sistema de telefonia visando compreender este um pouco mais a fundo,

muitas vezes objetivando obter vantagens indevidas (por exemplo: realizar chamadas telefônicas gratuitas e/ou internacionais, acessar lugares restritos da rede telefônica, descobrir dados de um usuário de telefonia) [9]. Muitos historiadores consideram os *phone phreakers* os precursores da cultura *hacker* [9].

Destarte, anos após o conceito de Engenharia Social ter sido formulado, ele teve o seu uso ressignificado, pois através da assimetria epistêmica entre as vítimas da engenharia social e os atacantes, e a dominância tecnocrática desses, os engenheiros sociais conseguem realizar uma substituição teleológica que seja favorável a seus objetivos [9].

Um dos mais notáveis engenheiros sociais foi o *hacker* Kevin Mitnick, que no seu livro "A arte de enganar"[3] conta várias histórias exemplificando como um atacante pode usar a engenharia social para obter vantagens indevidas, sempre tomando o cuidado de explicar quais fatores influenciaram para o sucesso do golpe, com o objetivo de alertar como os atacantes agem no dia-a-dia para enganar suas vítimas.

2.3 Detalhando a ameaça do *phishing*

Figura 2.1: Ilustração de um ataque de *phishing*. Fonte: *Pixabay*, sob licença *creative commons*.

A figura 2.1 ilustra um ataque de *phishing*. O ninja com a vara de pescar representa o atacante lançando a isca (representada pelo anzol) para a vítima, encarnada pelo usuário do computador. Apesar das definições citadas anteriormente na seção 1.3 deste trabalho abordarem o *phishing* como um mecanismo para o roubo de informações, essa técnica

pode ser utilizada simplesmente para obter algum tipo de vantagem indevida sobre o alvo, por exemplo o *phishing* pode servir como um vetor de disseminação de *malware*.

A primeira tentativa conhecida de golpe de *phishing* aconteceu em 1996 [24], e em 2006 os ataques dessa natureza contra instituições financeiras cresceram cerca de 10% [24]. Hoje, mais de 20 anos após a primeira tentativa de ataque de *phishing*, o *phishing* continua sendo uma das formas mais comuns de golpe digital, de acordo com o relatório anual de segurança cibernética da *Cisco Systems* referente ao ano de 2017 [25]. No relatório, o *phishing* figurava na 7ª colocação entre os empregos de *malware* [1].

O *phishing* é uma técnica que pode ser usada como estágio em um ataque, por exemplo uma *APT* (*Advanced Persistent Threat*) [10, 26]. Acredita-se que isso ocorra porque é um método simples e de baixo custo [10, 15]. Não há expectativas para que esse tipo de ataque caia em desuso. Pelo contrário, acredita-se que ele fique cada vez mais sofisticado [15]. Por isso é muito importante saber o que torna um ataque de *phishing* bem sucedido. Apesar disso, existem poucos trabalhos desenvolvidos no sentido de identificar critérios de "sucesso" para um ataque de *phishing* [12]. Também há uma carência de literatura científica sobre métodos "não convencionais" de *phishing*, por exemplo métodos que utilizam como vetores (instrumentos usados para realizar ataques) diferentes de ligações telefônicas, correio eletrônico ou unidades USB [13].

2.4 Automatização de ataques de engenharia social por *Bots*

Ataques de engenharia social automatizada representam uma séria ameaça à segurança da informação [14]. Isso acontece porque esses exploram a engenharia social, que como já foi discutido previamente(seções 1.2 e 2.2) é uma ferramenta muito poderosa, pois explora o "elo mais fraco da segurança da informação", mas fazem essa exploração de maneira automatizada aumentando o alcance dos golpes [14] e tornando-os mais baratos [7]. Esta seção objetiva mostrar mais sobre a estrutura e o funcionamento desse tipo de sistema.

Na literatura, podemos citar algumas referências de ataque de engenharia social automatizada. Para a concepção deste trabalho podemos citar os trabalhos de [13], que fez o uso de *BOTs* que simulam o comportamento humano para testes sobre *phishing* no twitter; de [14] que faz um ataque simulando o comportamento humano no *IRC*; [7] no qual é executado um ataque de engenharia social automatiza na rede social *facebook*; e em [11] onde o alvo é um site de leilões online.

[1] *Malware* é qualquer tipo de artefato computacional que tem por objetivo realizar um ataque virtual a pessoas e (ou) computadores) observados com mais frequência.

Em [7], além dos testes com ataques de engenharia social automatizada realizados, é importante destacar como um dos resultados impactantes atingidos pelos pesquisadores foi definição de um ciclo no qual a maioria dos ataques de engenharia social automatizada (através de *BOTs*) se encaixa. Este ciclo é composto de 5 etapas que são descritas a seguir:

1. Planejamento: O atacante define como o *BOT* irá se comportar durante o ataque, quais informações são importantes para identificar alvos para o ataque, e o que ele fará depois que o ataque se concretiza;
2. Mapeamento de alvos: O *BOT* realiza automaticamente o mapeamento dos alvos do ataque, pegando as informações necessárias definidas no passo anterior.
3. Execução: O *BOT* executa o ataque, conforme definido no planejamento. Podendo ser um robô que simula o comportamento humano e pede informações, um ataque de *phishing*,ou apresenta um link para um *malware*.
4. Recrutamento ou camuflagem: Caso o recrutamento[2] seja a estratégia adotada pode-se utilizar as vítimas do ataque para disseminar mais ainda o ataque e/ou manter contato com a vítima para uma nova requisição de informações no futuro. Caso a camuflagem[3] seja a escolhida o *BOT* deve esconder o ataque, excluindo quaisquer vestígios deixados.
5. Evolução ou regressão: O sucesso do *BOT* é mensurado, e caso ele tenha sido bem sucedido, são analisados quais os fatores que influenciaram no sucesso, para que estes sejam adotados em ataques futuros. Caso o ataque não tenha obtido sucesso, o *BOT* regressa para um ataque mais simples, procurando outra estratégia.

2.5 *Twitter*

O *twitter* consiste em um microblog, onde os usuários postam mensagens curtas (doravante chamadas apenas como tuítes ou *tweets*), com tamanho de até 280 caracteres textuais. Os tuítes podem conter alguns elementos multimídia como, por exemplo: vídeos, imagens, enquetes e/ou geolocalização.

Contas no *twitter* se relacionam seguindo umas as outras. Cada conta possui uma linha do tempo, que é a página principal da rede social, onde são mostrados todos os tuítes das contas que são seguidas pelo usuário. Além disso o *twitter* permite a escrita de um tuíte mencionando uma outra conta. Para fazer isto basta escrever o caractere '@'(arroba)

[2]Ato de reunir, alistar ou convocar pessoas para um determinado propósito
[3]Arte de dissimular tropas e materiais bélicos à observação inimiga.

seguido pelo *username* do usuário. Com isso, toda vez que uma conta é mencionada o usuário recebe uma notificação. Outro artifício ao qual duas contas podem utilizar para se comunicar, é o mecanismo de mensagem direta, onde as duas contas podem trocar mensagens em um *chat*, de maneira totalmente privada.

O *twitter* também faz um mapeamento dos 'assuntos do momento'. Estes são os assuntos mais comentados no momento, e o usuário pode escolher como deseja segmentar esses assuntos: por uma região específica, ou no mundo inteiro. Para facilitar a filtragem dos assuntos do momento pela mídia social existem as *hashtags*. *Hashtags* nada mais são que palavras-chave marcadas pelos usuários através de um caractere especial. Para utilizar o recurso basta escrever o caractere '#'(cerquilha) seguido pela palavra-chave desejada. O uso de *hashtags* é importante para impulsionar a publicação, desta forma, mais usuários podem ver as publicações marcadas ao pesquisar pela *hashtag* na plataforma [27].

2.5.1 Coletando dados da mídia social Twitter

Conforme descrito na seção 1.4 o twitter é uma mídia social que funciona no modelo de microblog (uma forma de blog, onde os usuários postam mensagens muito curtas para visualização por meio de uma rede de pessoas). Existem 4 (quatro) meios de fazer o uso das funcionalidades da plataforma, que são: uso do site, uso de aplicativo, uso de *API REST* e incorporação em sítios de terceiros.

Através das APIs, qualquer desenvolvedor que queira utilizar a plataforma (seja para utilizar os dados gerados pelos usuários do twitter ou para criar novos dados dentro da plataforma) pode cadastrar o aplicativo no site Twitter Apps e após a aprovação do aplicativo pelo twitter, será fornecido o acesso à *API*.

O twitter possui 4 APIs elas são:

- *Standard APIs*: São as APIs gratuitas que são liberadas para os desenvolvedores criarem aplicativos em cima da plataforma.
- *Premium APIs*: São as APIs pagas para os desenvolvedores criarem aplicativos em cima da plataforma, possuem maior escalabilidade em seu uso.
- *Enterprise APIs*: São as APIs pagas para os desenvolvedores criarem aplicativos em cima da plataforma, são as mais completas comparado com as duas anteriores.
- *Ads APIs*: São as APIs que são voltadas para a criação de propagandas dentro da plataforma.

Para realização do experimento aqui relatado foi utilizada a *Standard API* pois ela é gratuita, ao contrário das *APIs Premium* e *Enterprise*. Já a *Ads API* é focada na

criação de anúncios na plataforma, o que não foi o foco desta pesquisa científica. Por isso, doravante sempre que neste documento constar uma menção à *API* do *twitter* ou a *API REST* do *twitter*, fica subentendido que está sendo discutido o uso da *Standard API*.

2.5.2 O que é uma *API REST*?

Existem dois conceitos distintos que compõe o que é uma *API REST*, pois esta é uma *API* que utiliza a arquitetura *REST*. Portanto, para entender a definição de *API REST* é preciso compreender o que é uma *API* e o que é a arquitetura *REST*.

API é uma sigla para *Application Programming Interface*, como o próprio nome sugere é uma interface provida por um serviço, na qual através desta outras aplicações podem acessar recursos deste. Um exemplo famoso de *API* é a *API Win32* do *Windows* que provê acesso aos programadores às chamadas do sistema operacional *Windows*, em todas as suas versões, desde o *Windows 95* [1, p. 36].

Por sua vez, *REST* é uma abreviação para *Representational State Transfer*. O termo foi criado em [28]. É uma arquitetura para desenvolvimento para sistemas distribuídos, que não foca em detalhes da implementação do e foca nas regras dos componentes, nas regras de interação entre os componentes e na interpretação do significado dos elementos de dados [28]. Serviços web que utilizam a arquitetura *REST* provém uma semântica de interfaces homogênea baseada no protocolo *HTTP* [29], facilitando a interoperabilidade entre sistemas e permitindo que sistemas manipulem recursos da *web*.

2.5.3 Acessando a *API* do *Twitter* em *Python*

Vislumbrando o acesso às funcionalidades da plataforma, a *API* do twitter disponibiliza uma série de métodos, segundo o padrão *REST*. A lista completa destes pode ser acessada no índice de referências da *API* [30].

Para utilizar os métodos disponíveis é possível fazer requisições *HTTP* diretamente à plataforma seguindo a documentação oficial disponibilizada pelo twitter. Entretanto, para facilitar existem diversas bibliotecas que encapsulam as chamadas *HTTP*, com o objetivo de facilitar o trabalho dos desenvolvedores.

Uma destas bibliotecas, na linguagem *Python*, é a *tweepy* possui o código aberto sob a licença *MIT*. Essa biblioteca que provê o acesso grande parte dos métodos da *API* do twitter de maneira mais descomplicada (se comparado a realizar as requisições *HTTP* manualmente), e por isso foi adotada para a realização dos experimentos. A lista completa dos métodos que são cobertos pela *tweepy* estão em [30].

2.5.4 *Twitter APP - TAPP*

Segundo as regras do *twitter*, toda e qualquer aplicação que envolva comunicação com a *API REST* da organização precisa estar cadastrada na base de dados desta empresa. Dentro da plataforma a aplicação é chamadas de *Twitter APP* (doravante nomeadas apenas como *TAPP*)

Destarte, existe um maior controle sobre as aplicações, para que estas não entrem em prováveis conflitos de interesse com a plataforma [31]. Para cadastrar uma aplicação basta acessar a página *web* do *twitter* para desenvolvedores e seguir as recomendações lá sugeridas.

Ao cadastrar a aplicação, o *twitter*, fornece ao usuário 4 chaves nomeadas: *consumer token, consumer secret, access key* e *access secret*. Todas estas chaves são utilizadas na autenticação com a API.

2.5.5 Passos para a utilização de um *TAPP*

Esta subseção discorre sobre a sequência de etapas seguidas no experimento, para o desenvolvimento de um *TAPP* funcional. Acredita-se que grande parte dessas etapas são comuns na criação de qualquer *TAPP* genérico

Registrando-se como usuário da *API*

O primeiro passo para a criação de um *TAPP* (não importando qual seja a sua funcionalidade) é se registrar como um usuário da *API* perante o *twitter*, conforme foi brevemente citado na seção 2.5.4.

Primeiramente, para se registrar como um usuário da *API*, é necessário possuir uma conta do twitter. Em posse de uma conta do twitter basta acessar a página web `https://apps.twitter.com/` e solicitar uma conta de desenvolvedor à plataforma. Após a solicitação o *twitter*, irá analisar o pedido, podendo conceder ou não a conta de desenvolvedor ao solicitante.

Na época em que foram criadas as contas utilizadas nesta pesquisa, as regras do twitter para criação de *TAPPs* eram mais brandas. Bastava o usuário possuir uma conta no twitter e ao solicitar as chaves de acesso lhe eram disponibilizadas automaticamente, sem nenhuma análise.

As regras antigas facilitaram o uso de 5 contas distintas para a realização deste experimento. Entretanto, como consta na página `https://apps.twitter.com/` em julho de 2018 aconteceram mudanças nas regras. , devido à entrada em vigor da *GDPR* na União Europeia [32].

Autenticando-se com a *API*

Antes de realizar qualquer operação com a *API* do *twitter* é necessário realizar uma autenticação, utilizando as chaves de acesso obtidas no cadastro do *TAPP*. No algoritmo 2.1 é mostrado como realizar o procedimento de autenticação utilizando a biblioteca *tweepy*. Após a autenticação também é mostrado ao usuário a sua linha do tempo.

```
1  import tweepy
2
3  \#setar as credenciais de acesso
4  auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
5  auth.set_access_token(access_token, access_token_secret)
6
7  \#autenticar
8  api = tweepy.API(auth)
9
10 \#recuperar uma lista de tu tes da plataforma
11 public\_tweets = api.home_timeline()
12 \#imprimir os tu tes da lista
13 for tweet in public_tweets:
14     print tweet.text
```

Algoritmo 2.1: 'Hello tweepy'. Fonte: Documentação oficial do *tweepy*

O algoritmo 2.1 é um *Snippet* que mostra como autenticar com o twitter e mostrar a *timeline* do usuário possuidor das chaves de acesso. Analisando o algoritmo é fácil perceber a simplicidade oferecida pela biblioteca adotada. Em poucas linhas de código é possível setar as credenciais de acesso(linhas 4 e 5), se autenticar com a *API*(linha 8) e obter dados da plataforma em tempo real(linha 11). A *API* do *twitter* sempre retorna objetos de dados no formato *JSON*. Os atributos destes objetos serão melhor detalhados nas seções 2.6.1 e 2.6.2 deste trabalho.

Consumindo dados

Existem diversos dados que podem ser consumidos através da *API* do twitter. Tais como dados sobre usuários, geolocalização, tuítes, *trending tópics* e etc. Este trabalho focou nos dados de tuítes, pois já possuem os dados do autor embutidos no seu *JSON* (seções 2.6.1 e 2.6.2).

Exitem 2(dois) recursos aos quais desenvolvedores podem recorrer ao tentar obter dados sobre tuítes, baseando-se em palavras-chave. O primeiro é o uso do método `search` da *API*. Este retorna uma lista de até 1500 tuítes, realizados 2(duas) semanas antes da utilização do método. As opções de retorno do método podem ser ajustadas através dos parâmetros. Uma explicação mais detalhada sobre o método encontra-se em [33], onde são

detalhados todos os métodos da biblioteca *tweepy*. O ponto negativo do uso do método `search` é que a lista de tuítes que ele retorna não é em tempo real, a lista contém tuítes postados até duas semanas atrás a partir do momento em que a requisição foi feita.

Outra forma para recuperar tuítes é utilizando uma *stream*. Ao utilizar uma *stream* os tuítes são recuperados em tempo real, a partir do momento que esta foi inicializada. A utilização de uma *stream* é bem simples e a documentação oficial do *tweepy* possui um tutorial, que mostra um passo-a-passo de como utilizar este recurso de maneira bem simples e objetiva. O código de como montar uma *stream* básica pode ser consultado no algoritmo 2.2 que mostra como montar uma *stream* que retorna todos os tuítes que contem a palavra-chave *python*.

```
import tweepy
\#sobrescreve a classe tweepy.StreamListener para acrescentar l gica ao
    m todo on\_status
class MyStreamListener(tweepy.StreamListener):
    def on_status(self, status):
        print(status.text)

myStreamListener = MyStreamListener()
myStream = tweepy.Stream(auth = api.auth, listener=myStreamListener())

myStream.filter(track=['python'])
```

Algoritmo 2.2: *Snippet* de uma *stream*. Fonte: *Streaming With Tweepy* [34]

Estimulando interações por meio de postagens com robôs

Para cumprir os objetivos propostos neste trabalho, foi necessário conceber um software que fosse capaz de realizar pseudo-ataques de *phishing*. Conforme foi discutido previamente, para isso é necessária a interação da vítima com uma isca enviada. Como o vetor para o pseudo-ataque de *phishing* escolhido é o twitter, é necessário que os participantes do experimento de alguma maneira entrem em contato com a isca.

Conforme foi explicitado na seção 2.5, existem dois métodos nos quais dois usuários podem se comunicar entre si dentro da plataforma: As menções e as mensagens diretas. Ambos os métodos possuem suporte para o uso através da API, e poderiam ser utilizados como vetores para as iscas. Entretanto, como as mensagens diretas são privadas e podem ser consideradas mais íntimas, o vetor escolhido para a realização dos testes foi as menções.

Para realizar as postagens, foi utilizado o método `update_status`, que é o método que realiza a postagem de tuítes na biblioteca *tweepy* (para detalhes, vide[33]).

2.6 Objetos da *API* do *Twitter*

Nesta seção são discutidos alguns conceitos que foram importantes na concepção do software utilizado nesta pesquisa. Quando requisitada, a *API* do twitter retorna objetos no formato *JSON*. Existem padrões aos quais esses objetos seguem. Serão abordados os principais objetos retornados pela *API*, além de outros atributos que podem ser derivados da conta.

2.6.1 A estrutura de dados de um tuíte

Um tuíte possui diversos atributos, dentre esses alguns elementos merecem ser destacados. Por exemplo, em um tuíte (figura 2.2) há o apelido do autor, o texto deste, a data que o tuíte foi publicado, as interações que o tuíte recebeu (curtidas e retuítes) e etc. Conforme já foi citado na seção 2.5, um tuíte pode conter diversos tipos de informação.

Figura 2.2: Um tuíte.

Devido ao fato do tuíte possuir várias características diferentes, e aliado à necessidade do objeto JSON refletir de maneira fidedigna o tuíte, é natural que o JSON possua vários atributos, conforme pode ser visto na figura 2.3. A lista completa dos atributos encontra-se em [35].

Segundo [35], o objeto JSON referente ao tuíte possui uma longa lista de atributos incluindo alguns atributos fundamentais tais como: `id` o identificador único do tuíte dentro da plataforma, `created_at` a data de criação do tuíte, e `text` o texto de 280 caracteres do tuíte. Ainda segundo [35] um tuíte possui ainda vários objetos filhos. Os objetos filhos incluem representações em JSON do usuário que postou o tuíte e/ou de

```
{
 "created_at":"Thu Apr 06 15:24:15 +0000 2017",
 "id": 850006245121695744,
 "id_str": "850006245121695744",
 "text": "1/ Today we're sharing our vision for the future of the Twitter API
 platform!nhttps://t.co/XweGngmxlP",
 "user": {},
 "entities": {}
}
```

Figura 2.3: O objeto JSON de um tuíte. Fonte: *Twitter for developers*.

recursos multimídia. Na figura 2.3 são mostrados apenas os atributos em nível de raiz, os atributos filhos são representados apenas com '', vislumbrando a tornar mais simples e legível.

2.6.2 A estrutura de dados de uma conta no Twitter

O objeto JSON de representação de uma conta no twitter analogamente ao tuíte, também possui uma série de atributos. Podemos ver na figura 2.4 alguns destes atributos tais como aimegem do perfil, o nome do usuário, o apelido do usuário, o número de seguidores, o número de seguidos, a descrição da conta e etc.

Figura 2.4: Uma conta. Fonte: *Twitter for developers*.

Esse objeto contém os metadados públicos que descrevem esta conta [36]. Segundo [36] existem alguns atributos sobre as contas que não costumam mudar, por exemplo o código identificador e a data em que a conta foi criada. Há também atributos que não mudam com tanta frequência (como as informações que o usuário opta por mostrar no seu perfil) e atributos que mudam com maior frequência. Um exemplo de atributo de mudança fraquente é é o número de tuítes que a conta postou. A lista completa dos atributos encontra-se em [36].

Figura 2.5: O objeto JSON referente a um usuário. Fonte: *Twitter for developers*.

2.6.3 Autoridade: Um atributo derivado de contas no *Twitter*

No artigo '*Ranking approaches for microblog search*' [20] são experimentadas abordagens para o ranqueamento de tuítes em buscas em tempo real. Conforme citado em 1.4 este trabalho baseou-se nesse artigo para o ranqueamento de contas.

Em [20] definiu-se modelos matemáticos para mensurar a autoridade do autor do tuíte de um tuíte, que são o *TweetRank* e *FollowerRank*. Baseando-se nestes foi concebido um arcabouço para mensurar a importância de um tuíte. Os valores desses atributos foram avaliados por usuários humanos, verificando se os tuítes considerados relevantes pelas pessoas coincidiam com os apontados pela metodologia.

Intuitivamente, se um usuário tem muitos seguidores, quer dizer que ele está propagando informação útil. Ou se alguém está propagando informação útil muitas, pessoas irão seguir esse alguém. Por isso, foi concebido o *FollowerRank*, que é interpretado como a autoridade que a conta tem. Este é descrito pela fórmula 2.1, onde *i(a)* é o número de seguidores que a conta possui (`followers_count`) e *o(a)* é o número de contas que a conta segue na plataforma (`friends_count`) [20].

$$FR(a) = \frac{i(a)}{i(a) + o(a)} \tag{2.1}$$

O modelo criado por [20] foi validado por usuários humanos em um experimento que pode ser descrito da seguinte forma:

1. Foram mostrados vários tuítes para os participantes da pesquisa;
2. Os participantes tinham que classificar os tuítes que eles achavam mais relevantes;
3. Foram comparados os tuítes que os participantes consideravam relevantes com os tuítes que a metodologia apontava como relevantes.

Baseando-se no experimento, [20] constataram que a abordagem que levava em consideração o *FollowerRank* e alguns atributos do tuíte (tais como o tamanho do tuíte e/ou a presença de um link) apresentou resultados melhores, e que apesar da metodologia precisar de algumas melhorias, ela é aplicável no ranqueamento de tuítes em tempo real. Seria a autoridade de uma conta no twitter um indicador de propensão dos seus usuários a compartilhar dados com estranhos?

2.7 Regressão logística

Ao estudar fenômenos, é comum elaborar modelos que permitem compreender e testar mais hipóteses sobre esses. O uso de regressões para obter fórmulas matemáticas que descrevem o comportamento de fenômenos é uma das técnicas podem ser adotadas em determinados contextos. Dentre os modelos de regressão existentes este trabalho adota a regressão logística. O uso desse tipo de regressão ocorreu pois acredita-se que o fenômeno observado seria bem descrito através desta ferramenta, e além disso, dentro desse domínio de aplicação encontrou-se na literatura um trabalho relevante [10] que utiliza esta técnica estatística.

Diferentemente da regressão linear, uma regressão logística é utilizada quando deseja-se tentar prever os possíveis valores assumidos por uma variável dependente é categórica, através de variáveis independentes que podem assumir qualquer forma (categórica ou não) [37, 38]. A função que descreve a relação entre as variáveis independentes e a variável dependente em uma regressão logística é a da fórmula 2.2 [38].

$$\Theta = \frac{e^{(\alpha+\beta_1 x_1+\beta_2 x_2+...+\beta_i x_i)}}{1+e^{(\alpha+\beta_1 x_1+\beta_2 x_2+...+\beta_i x_i)}} \tag{2.2}$$

Com base nos conceitos apresentados neste capítulo, foi concebida a metodologia apresentada no capítulo 3. Além disso, o conteúdo aqui apresentado serve como base para guiar a compreensão do resultados obtidos e mostrados no capítulo 4. E este capítulo serve como alicerce para a discussão, apresentada no capítulo 5.

Capítulo 3

Metodologia

Este capítulo apresenta o percurso metodológico seguido pelos pesquisadores, desde as primeiras ideias desenvolvidas, até a realização dos experimentos, problemas encontrados e alternativas construídas. A concepção da metodologia para a realização da pesquisa aqui relatada foi em grandes proporções baseada em 2 (dois) pressupostos, brevemente explicados abaixo.

1. Quem compartilha informação com mais facilidade (mais indiscreto) é mais vulnerável a golpes onde os atacantes solicitam por informações. Dentre esses golpes se encontra o *phishing*. Isso foi abordado conceitualmente no capítulo 2.
2. A disposição de alguém compartilhar informação com um interlocutor desconhecido aumenta na medida em que eles discutem um mesmo assunto de interesse. Isso também foi abordado conceitualmente no capítulo 2.

Baseando-se nesses pressupostos, conduziu-se uma pesquisa descritiva, exploratória e quantitativa, através de um experimento, buscando ratificar (ou falsear) a hipótese estabelecida e seus pressupostos (1.6). Primeiramente os pseudo-ataques de *phishing* foram realizados baseando-se no artigo escrito por Jakobsson[12], onde ele desenvolve uma discussão sobre como desenvolver uma pesquisa sobre este assunto de uma maneira ética. Foi construída uma sequencia de experimentos com graus de complexidade cada vez maiores, onde os dados e resultados obtidos em um experimento de sequência x indicavam os caminhos seguintes de funções para serem inseridas no experimento de sequencia x + 1, ou seja, usando um método indutivo. Foram realizandos quatro experimentos, numerados 1 a 4.

3.1 Organização dos Experimentos

Os 4 (quatro) experimentos de forma geram foram estruturados nos cinco passos a seguir descritos.

3.1.1 (1) Definição do objetivo do experimento

Cada um dos experimentos atendia de forma incremental ao alcance de um objetivo. O primeiro experimento vislumbrava apenas mapear como os usuários se comportavam na plataforma. Os demais experimentos objetivaram construir um arcabouço útil para execução de ataques de engenharia social automatizada, ocorrendo melhorias a cada iteração.

3.1.2 (2) Planejamento dos pseudo-ataques

O planejamento dos pseudo-ataques consistiu em definir o que seria executado para cumprir o objetivo anteriormente definido, e lançou mão da definição precisa do que seria feito em cada uma das seis sub-etapas genéricas, a seguir definidas.

Definição da fonte do fluxo A fonte do fluxo de tuítes que deveria ser obtida através do uso de palavras-chave ou de outras técnicas, para separar os possíveis alvos por grupos de interesse.

Obtenção do fluxo Deveria ser obtido do *twitter* um fluxo contínuo de tuítes associados aos grupos de interesse.

Amostragem Na medida em que o fluxo de tuítes fosse consumido, deveria realizar-se uma amostragem das contas.

Envio de Estímulos (iscas) Para as contas da amostragem, deveriam ser enviados estímulos ou iscas, na forma de novos tuítes, oferecendo informações de interesse.

Análise da resposta aos estímulos Para os usuários que "morderam a isca", lhes foi apresentada uma página que buscava coletar dados pessoais do usuário.

Contabilização Os dados que descrevem o comportamento do usuário seriam contabilizados.

3.1.3 (3) Implementação do software

Utilizando-se a linguagem *Python* , foram implementadas 3 versões distintas do software de ataques de engenharia social automáticos, além de uma versão preliminar que serviu como um estudo de distribuição dos usuários do twitter.

3.1.4 (4) Testes básicos

Para avaliar se o funcionamento do *software* estava coerente, foram realizados testes no programa de computador, sem que as mensagens (ataques e/ou tuítes simulando um usuário humano) fossem de fato enviadas aos usuários.

3.1.5 Análise dos dados

Com o término dos experimentos conduziu-se uma análise dos dados obtidos durante a pesquisa, como objetivo de encontrar alguma correlação de elementos que possam indicar a vulnerabilidade de uma conta no *twitter*.

3.2 Pós experimento

Após a realização dos experimentos foram realizados os seguintes passos.

Documentação das arquiteturas de código desenvolvido Análise final do código-fonte desenvolvido para este experimento, como foco de documentar o desenvolvimento com a finalidade de tornar o entendimento do programa de computador mais simples.

Produção da monografia Escrita do trabalho científico descrevendo os principais tópicos que permearam a condução da pesquisa.

3.3 Contas do *twitter* utilizadas no experimento

Como já informado no capítulo 2, segundo as regras do *twitter*, toda e qualquer aplicação que envolva comunicação com a *API REST*, precisa estar cadastrada na base de dados.

No meio da realização da pesquisa o Twitter passou a exercer maior controle sobre as aplicações, para que estas não entrem em prováveis conflitos de interesse com a plataforma. Para cadastrar a aplicação foi acessada a página web `https://apps.twitter.com/` e seguidas as recomendações lá sugeridas.

Com o término da implementação da primeira versão, que era capaz de realizar algum tipo de teste em *phishing*, se fez necessário o acesso à várias aplicações em contas diferentes. Com isso, notou-se que o twitter havia mudado as regras de utilização da *API*, tornando mais difícil o cadastro de novas aplicações. Entretanto, um pesquisador associado ao grupo de pesquisa Inteligência Cooperativa em Redes Sociais Complexas [39] já havia criado previamente contas e aplicações na plataforma e cedeu suas credenciais de acesso para realização desta pesquisa.

Como já informado no capítulo 2, ao cadastrar a aplicação, o twitter fornece ao usuário 4 chaves nomeadas: *consumer token, consumer secret, access key* e *access secret*. Todas estas chaves são utilizadas na autenticação com a API. Para este projeto foram utilizadas cinco contas do twitter, com uma aplicação cadastrada em cada uma, permitindo a atividade de vários robôs ao mesmo tempo. As contas foram:

- conta1: Experimentos 3 e 4.
- conta2: Experimentos 3 e 4.
- conta3: Testes obtenção de dados para o mapeamento dos usuários do *twitter* experimento 1, e execução do experimento 3.
- conta4: Testes obtenção de dados para o mapeamento dos usuários do *twitter* experimento 1, e execução do experimento 2.
- conta5: Testes obtenção de dados para o mapeamento dos usuários do *twitter* experimento 1, e execução do experimento 2.

Na época em que as aplicações foram cadastradas, o *twitter* possuía regras mais flexíveis para a criação de ferramentas que se comunicam com a *API*, o que possibilitou a criação de vários *BOTS*. Nas regras atuais, a criação de vários robôs seria uma tarefa muito mais árdua, pois cada aplicação deve ser submetida a um processo mais rigoroso de aprovação pelo *twitter* [31].

3.4 *Software*

A arquitetura de todos os robôs desenvolvidos é apresentada no capítulo de resultados e se refere tanto à estrutura dos softwares como também seus comportamentos.

Todos os códigos foram desenvolvidos com abordagem de código aberto, sendo escritos sob licença *GPL* versão 3, e encontram-se hospedados no *github*, podendo ser acessados através do link `https://github.com/jeffvfa/bot-python/`, sujeita a autorização do autor e de seu orientador.

Todos os softwares foram desenvolvidos entre o período de 2, 1 de agosto de 2018 a 12 de novembro de 2018. Segundo o *WakaTime*[1] foram gastas 42 horas e 30 minutos apenas com programação, para a produção total de 1796 linhas de código no experimento 5. O esforço total para desenvolvimento foi estimado, grosseiramente, em 90 horas a partir de análise de *logs*, *emails*, WakaTime, mensagens do *whatsapp* e arquivos drive trocados entre os pesquisadores.

[1]Serviço que registra métricas sobre o desenvolvimento de código-fonte através de *plugins open source*

3.5 Sistemas computacionais

Todos os programas foram executados usando dois computadores e um serviço que hospeda programas *python* na nuvem chamado *pythonanywhere*. Os robôs executaram em uma rede particular de um dos pesquisadores. A conexão utilizada foi wireless com velocidade de aproximadamente 31 *Mbps*.

O computador 1 foi o responsável por executar os BOTs, enquanto estes realizavam os experimentos. Este possui a seguinte configuração:

- Memória *RAM*: 7,8 *GiB*
- Processador: Intel® Core™ i5-2467M *CPU* @ 1.60GHz x4
- Placa de vídeo: Intel® Sandybridge Mobile x86/MMX/SSE2
- HD: 206,9 GB
- Sistema operacional: Ubuntu 16.04 LTS (32 bits)

O computador 2 foi mais utilizado no desenvolvimento do *software*, e na realização dos testes básicos. Este possui a seguinte configuração:

- Memória *RAM*: 7,8 *GiB*
- Processador: Intel® Core™ i7-3610QM *CPU* @ 2.30GHz x 4
- Placa de vídeo: NVIDIA® GeForce® GT 630M com 2GB DDR3 VRAM
- *HD*: 1 *TB*
- Sistema operacional: Ubuntu 16.04 LTS (64 bits)

Com base na metodologia descrita neste capítulo, foram conduzidos os experimentos que serão relatados no capítulo 4. No capítulo 5 se encontra a análise e discussão dos dados obtidos durante a experimentação. No capítulo 6 acha-se a conclusão.

Capítulo 4

Resultados

Este capítulo faz uma apresentação detalhada dos resultados obtidos com os quatro experimentos que visavam verificar a hipótese de que um dos fatores que pode influenciar a maior ou menor efetividade de um ataque de *phishing* no *Twitter*, provocando pelo envio de tuítes, é a autoridade do atacante. (seção 1.6). As diferentes arquiteturas de sistema desenvolvidas geraram distintos problemas, e também distintas soluções e distintos conjuntos de dados, para cada problema. No final de cada seção há uma breve discussão acerca do aprendizado obtido.

O experimento 1 visava obter familiaridade com os tipos de informações providas pelo twitter. O experimento 2 visava desenvolver uma implementação inicial de pseudo-ataques. O terceiro experimentou visou o desenvolvimento de uma estratégia mais complexa para pseudo-ataques de *phishing*. O experimento 4 visava a criação de um modelo mais sustentável para a realização de pseudo-ataques. Os objetivos de cada experimento não foram definidos a priori, isso é, foram definidos de forma incremental e exploratória, conforme os resultados do experimento anterior.

4.1 Experimento 1 - Iniciando a coleta

O experimento 1 foi i primeiro trabalho dos autores com a coleta de dados no twitter. Esse não tinha como objetivo realizar testes com *phishing* mas sim mapear características da população de usuários do twitter.

Utilizaram-se 2 (duas) contas do twitter, conforme citado na seção 3.3. Foram as contas cujo apelidos eram conta4 e conta5. O experimento foi estruturado nos seguintes passos:

1. Foi obtido do twitter um fluxo contínuo de tuítes associados a um conjunto de termos que representam assuntos em política, esportes ou entretenimento;

2. Na medida em que o fluxo de tuítes foi produzido, foi realizada a captura completa do objeto *JSON* retornado pela *API* do twitter.

3. Foram analisados os dados obtidos no passo anterior, buscando obter maiores características sobre como os usuários se distribuem na rede.

4.1.1 Escolha das áreas temáticas

O twitter é uma rede social bastante plural. Isso acontece devido ao fato desta mídia social possuir um grande número de usuários, onde cada um pode escrever postagens sobre os mais diversos assuntos. Devido à circunstância desse estudo objetivar representar o recorte mais preciso possível dos usuários vulneráveis, a escolha das áreas temáticas foi uma etapa importante para o desenvolvimento do experimento.

No final de cada ano, o twitter costuma publicar um relatório contendo um resumo das atividades que os usuários realizaram durante o ano na rede. Lendo textos jornalísticos [40, 41] que condensam estes relatórios com informações foco do Brasil é possível ter uma maior noção sobre os assuntos que foram mais comentados na rede. Os tópicos mais comentados se distribuem entre eventos musicais, *realitiy* shows, grandes eventos esportivos e sobre um conjunto de investigações da polícia federal brasileira que tinha como alvo principal políticos [40, 41]. As seguintes *hashtags* foram as mais comentados no Brasil no ano de 2017 [41]:

- #RockinRio
- #LavaJato
- #neymarjr
- #libertadores2017
- #bbmas (Billboard Music Awards)
- #mpn (Meus Prêmios Nick)
- #CamilaCabello
- #PablloVittar
- #BBB17
- #GameofThrones

Dessa lista, é possível depreender que os assuntos mais comentados pertencem a 3 grandes áreas: Entretenimento, esportes e política. Portanto, para viabilizar o estudo

Área temática	Palavras-chave
Política	'#politica', '#eleições', '#eleições2018', '#mulhernapolitica', '#mulheresnapolitica', '#jairbolsonaro', '#eleiçõesgerais', '#politicabrasileira', '#eleiçõessemlulaéfraude', '#pt', '#politicadobem', '#bolsonaro', '#aseleiçõesestãochegando', '#somostodosbolsonaro', '#bandeleições', '#bolsonaro2018', '#eleiçõesparaná2018', '#somosoutrapolitica', '#votobolsonaro17', '#psl', '#lula', '#mulheresnegrasnapolitica', '#estoucombolsonaro', '#política', '#politicatransparente', '#honesto', '#lulalivre', '#politica2018', '#forçabolsonaro', '#camaradosdeputados'
Esportes	'#esporte', '#futebol', '#esportes', '#futebolbrasileiro', '#esporteevida', '#copadomundo', '#futebolamericano', '#esporteinterativo', '#futebolarte', '#sport', '#globoesporte', '#futebolfeminino', '#esporteando', '#futebolamador', '#jogadordefutebol', '#corrida', '#copa2018', '#festafutebol', '#neymar', '#futebolsociety', '#esporteévida', '#flamengo', '#camisasdefutebol', '#copa', '#futebolamericanonobrasil', '#esportesradicais', '#museudofutebol', '#copadomundo2018', '#santosfutebolclube', '#esporteradical'
Entretenimento	'#entretenimento', '#música', '#filme', '#pop', '#music', '#músicaboa', '#cinema', '#filmes', '#kpop', '#arte', '#músicabrasileira', '#movie', '#rock', '#poprock', '#show', '#músicas', '#film', '#musica', '#culturapopular', '#músicasertaneja', '#popmusic', '#eventos', '#cultura', '#cine', '#filmedeterror', '#músicapopularbrasileira', '#art', '#popular', '#humor', '#músicaémix'

Tabela 4.1: *Keywords* geradas pelo *tagsfinder* para encontrar "vitimas"de pseudo-ataques de engenharia social.

do maior número de pessoas possível, essas foram as áreas temáticas escolhidas para o experimento.

Após a seleção das áreas temáticas, ocorreu a triagem de palavras chave. Objetivando uma escolha mais representativa e consistente, utilizou-se uma ferramenta de geração automática de *hashtags* (vide seção 1.4). A ferramenta eleita foi site *tagsfinder* , usada para geração das 30 *hashtags* mais populares em cada uma das três áreas temáticas. As *hashtags* geradas pelo *tagsfinder* foram utilizadas como palavras-chave na busca por "vítimas"para participar dos experimentos. A relação completa das palavras-chave utilizadas está descrita na tabela 4.1. Não há informação precisa sobre o dia exato em que as *hashtags* foram geradas, mas acredita-se que a datas de geração das *hashtags* é cerca de 2 a 3 dias antes do início do experimento 1. Julga-se que este conjunto de palavras-chave é suficientemente bom para a realização do experimentos.

4.1.2 Estrutura e funcionamento do experimento 1

Devido à sua simplicidade, o software desenvolvido para realização do experimento 1 possuiu uma arquitetura monolítica, ou seja, apresenta apenas um único módulo que

Área temática	Volume de dados
Política	10,5
Esportes	2,1 GB
Entretenimento	20 GB
Total	**32,6 GB**

Tabela 4.2: Distribuição da quantidade de dados por área temática

abarcava todas as suas funcionalidades. A figura 4.1 apresenta em uma máquina de estados a arquitetura dinâmica do experimento 1, isso é, seu funcionamento.

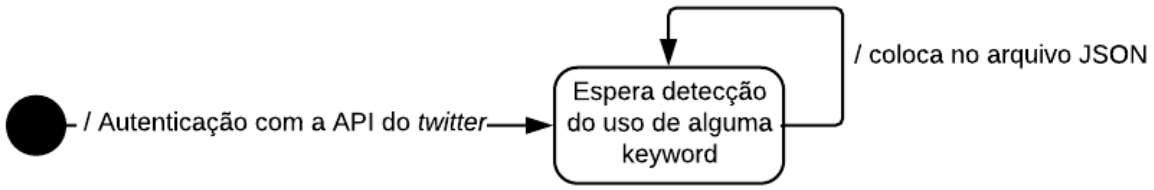

Figura 4.1: Máquina de estados que representa o comportamento do experimento 1.

4.1.3 Execução do experimento 1

A codificação do software para o experimento 1 foi feita ao longo de 3 semanas. A coleta de dados foi executado de 28 de setembro a 17 de outubro de 2018 de maneira ininterrupta. A codificação deste foi realizada no computador 2, e sua execução foi feita usando o computador 1 (ver seção 3.5).

4.1.4 Dados coletados do experimento 1

A autoridade calculada pelo *FollowerRank*(seção 2.6.3) é um número entre 0.0 e 1.0. Portanto, para este experimento foram consideradas 100 faixas de autoridade possíveis. Por exemplo, na faixa 0 são contas com a autoridade entre 0.000 e 0.009, a faixa 1 são contas com a autoridade entre 0.010 e 0.019 e assim sucessivamente. Foram adotadas 100 faixas neste experimento para ter um maior detalhamento da distribuição dos usuários dentro do *twitter*.

Durante a realização desse experimento foi coletada uma massa considerável de dados, contendo aproximadamente *32,6 GB* em arquivos no formato *JSON*, isso é, puramente textuais. A tabela 4.2 mostra a quantidade de dados coletados por área temática.

4.1.5 Análise preliminar do Experimento 1

Estes dados foram analisados em busca de identificação de padrões descritivos dos usuários do *twitter* que seriam as possíveis "vítimas"de pseudo-ataques. Para isso, foram calculadas as autoridades de cada autor de cada tuíte coletado, e foi realizada a distribuição.

A distribuição de frequência de tuítes que foram enviados por usuários numa faixa autoridade no tema política ocorreu segundo a figura 4.2. No eixo x estão representadas as 100 faixas de autoridade e no eixo y é mostrado a quantidade ou frequência de tuítes que foram enviados por usuários que possuem autoridade nessa faixa.

A figura 4.2 apresenta a distribuição de frequência de tuítes que foram enviados por usuários numa faixa de autoridade, para o tema política. Nota-se que os tuítes de maior frequência são aqueles emitidos por contas que possuem autoridade média, 0,5. Há, entretanto picos de frequência de envios de tuítes no extremo inferior e no canto superior das faixas, isso é, varias contas com autoridade muito baixa(sem seguidores), e várias contas com autoridade muito alta (quantidade de seguidores muito maior que a quantidade de seguidos) enviam muitos tuítes. Foram analisados 1405390 tuítes nesta distribuição.

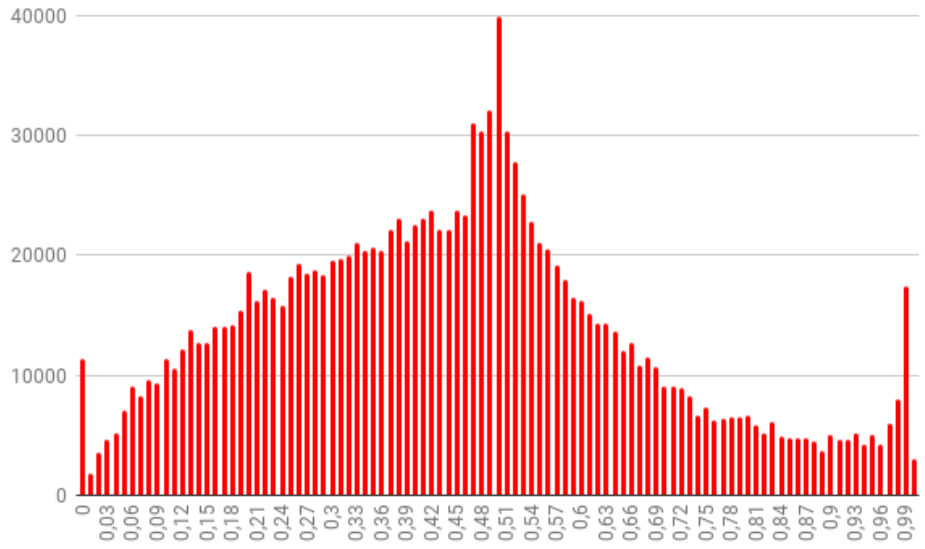

Figura 4.2: Distribuição de frequência de tuítes que foram enviados por usuários numa faixa autoridade, para o tema política.

A figura 4.3 apresenta a distribuição de frequência de tuítes que foram enviados por usuários numa faixa de autoridade, para o tema esportes. Nota-se que a distribuição é distinta da apresentada para o tema política. Embora continue a ter destaque os envios de tuítes no extremo inferior e no canto superior das faixas de autoridade, não se nota a clara

tendência à predominância de tuítes na faixa central de autoridade. Foram analisados 590728 tuítes nesta distribuição.

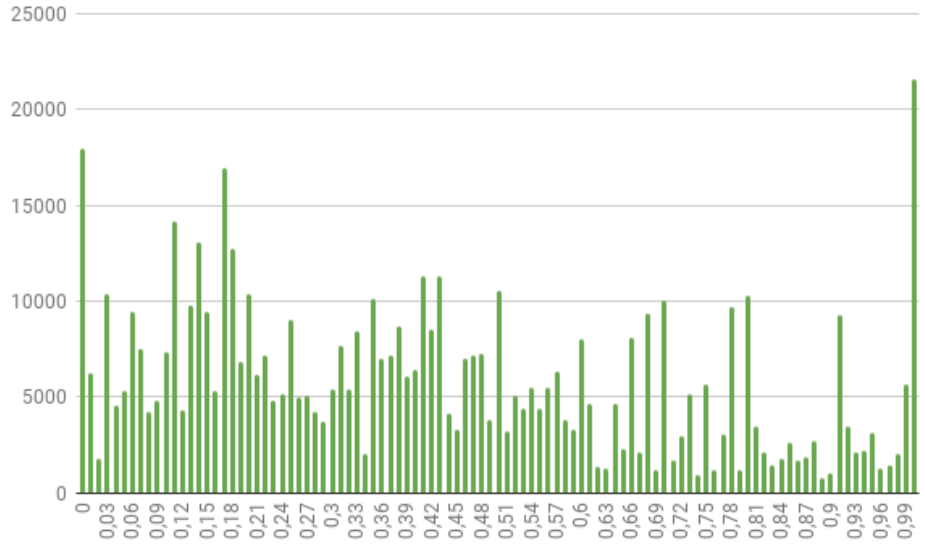

Figura 4.3: Distribuição de frequência de tuítes que foram enviados por usuários numa faixa autoridade, para o tema esportes..

A figura 4.4 apresenta a distribuição de frequência de tuítes por usuários numa faixa de autoridade, para o tema entretenimento. Nota-se que a distribuição é bem distinta da apresentada para o tema esportes, mas bem similar à apresentada nos tuítes sobre o tema política. Foram analisados 3159951 tuítes nesta distribuição.

A figura 4.5 apresenta a sobreposição das três curvas de distribuições de frequência de tuítes que foram enviados por usuários numa faixa de autoridade, para os três temas investigados. Nota-se que a quantidade de tuítes coletados no tema entretenimento foi praticamente o dobro da coletada no tema política,enquanto que foi bem menor a quantidade no tema esporte. De outra forma, observa-se similaridades entre as distribuições de frequência de política entretenimento.

4.1.6 Aprendizagem técnica com o experimento 1

Após a realização do experimento 1 foi compreendido melhor como a autoridade vinculada aos tuítes se distribui no *Twitter*. A análise dos dados permitiu a adoção de políticas de faixas de autoridade utilizadas nos próximos experimentos.

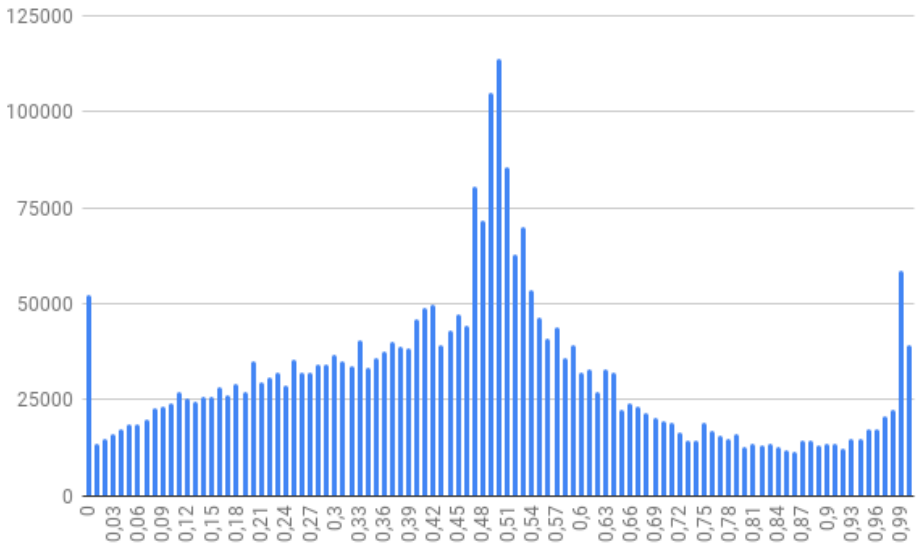

Figura 4.4: Contas x autoridade no assunto entretenimento.

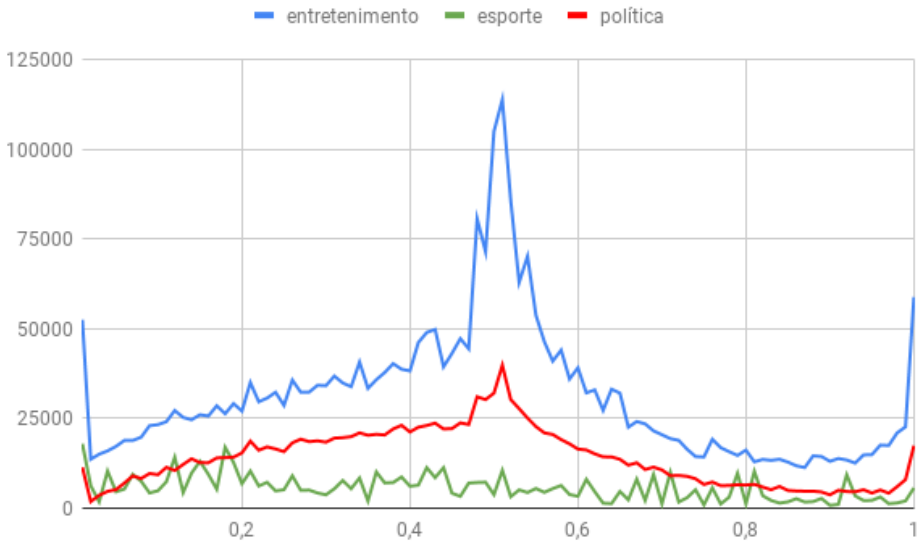

Figura 4.5: Linhas de contas x autoridade em todos os assuntos..

4.2 Experimento 2 - Implementação ingênua

O experimento 2 iniciou os ensaios com usuários reais ("vítimas"). O objetivo do segundo experimento foi testar a primeira estratégia de ataques de *phishing*. Foi nomeada, após o ocorrido, como implementação ingênua, pois consistia em usar robôs muito simples que somente procuravam alvos e enviava links de ataques por meio de menções aos usuários.

A figura 4.6 mostra alguns dos tuítes enviados no experimento 2. A observação dos tuítes deixa notável que os tuítes eram bem semelhantes.

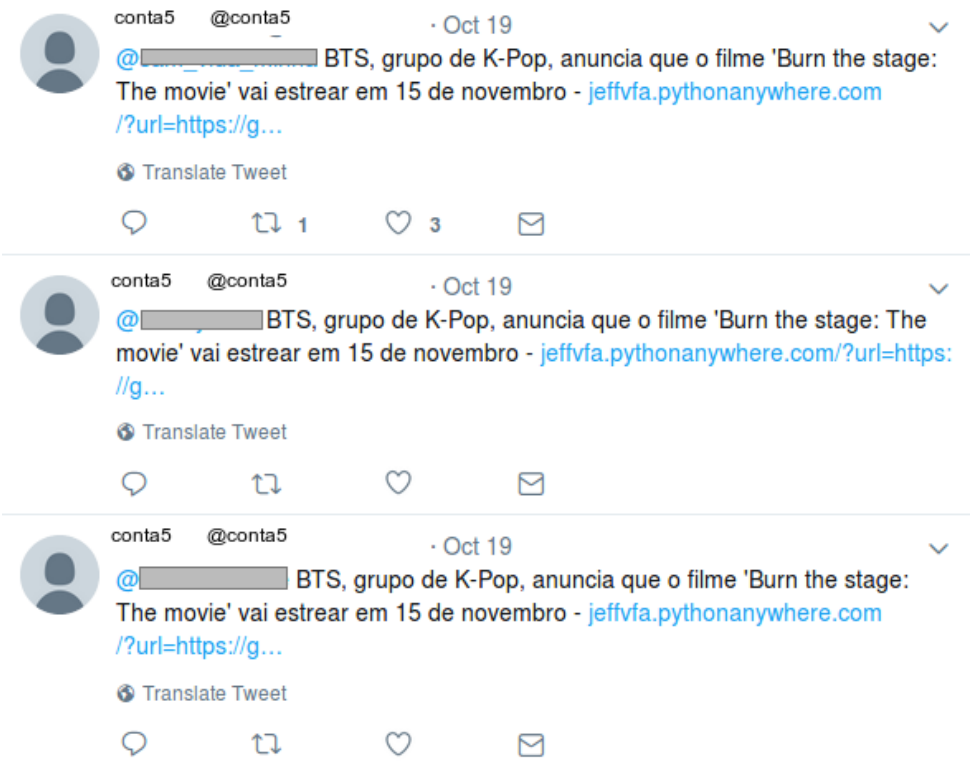

Figura 4.6: Alguns tuítes (iscas de pseudo-ataques de engenharia social) enviados no experimento 2.

O experimento 2 foi elaborado segundo trabalho do autor [12], vislumbrando a coleta de dados mais ética possível. O experimento foi estruturado nos seguintes passos:

1. Foi obtido do *twitter* um fluxo contínuo de tuítes associados a um conjunto de termos que representam assuntos em política, esportes ou entretenimento;
2. Na medida em que o fluxo de tuítes foi produzido, realizou-se uma amostragem das contas, com base na medida quantitativa da autoridade dessas contas;
3. Para as contas da amostra, foram enviados novos tuítes, oferecendo informações sobre temas de interesse associados ao assunto investigado, contendo um link que oferece informações legítimas sobre o assunto investigado;
4. Para os usuários que acessarem o link contido no tuíte enviado, antes do direcionamento para uma página com informação legítima, lhes foi apresentada uma página do sítio da pesquisa, que apresenta um formulário e solicita informação de dados pessoais do usuário, para um suposto cadastro. Esse acesso foi contabilizado. Na

página foram apresentados dois botões: “Acessar” e “Acessar sem cadastro”, e também um link pouco evidente, intitulado “Para ver o projeto desta pesquisa científica clique aqui”, cujo link direciona o usuário à leitura do projeto da pesquisa aqui relatada. O eventual acesso do usuário ao projeto de pesquisa foi contabilizado.

5. Ao clicar no botão de “Acessar sem cadastro”, independentemente do preenchimento de dados do formulário, o usuário foi redirecionado para uma página legítima que contém informações verdadeiras, abordando o assunto relacionado aos termos informados no passo 1. Foi contabilizado o eventual acesso do usuário à notícia, sem a realização do cadastro.

6. O botão “Cadastrar e acessar” foi acionável apenas caso tenham sido preenchidos dados não nulos em todos os campos do formulário. Do mesmo modo que ocorre com o pressionamento do botão “Acessar sem cadastro”, o usuário foi redirecionado para uma página legítima que contém informações verdadeiras, abordando o assunto relacionado aos termos informados no passo 1. O eventual acesso do usuário à notícia, com a suposta realização do cadastro, foi contabilizado. Os dados informados pelo usuário não foram enviados ao servidor, nem armazenados de qualquer forma.

7. Foram contabilizadas:(1) as contas amostradas segundo o uso das palavras-chave, para as quais foram enviados pseudo-ataques; (2) quais as autoridades dessas contas, obtidas por uma relação entre a quantidade de contas de seguidores e a quantidade de contas seguidas no *twitter*; (3) se estas contas acessaram o link enviado no pseudo-ataque; (4) Se o usuário da conta pseudo-atacada acessou o projeto da pesquisa; (5) se o usuário da conta pseudo-atacada acionou o botão “Cadastrar e acessar”, o que ocorre apenas se inseriu uma quantidade de caracteres não vazia nos campos que supostamente coletaram seus dados pessoais; (6) Se o usuário da conta pseudo-atacada acionou o botão “Acessar sem cadastro”. Nos casos 5 e 6, considera-se que foi apresentada a página que remete à informação de interesse do usuário. O caso 5 (preenchimento do formulário), sem que tenha havido o caso 4 (acesso ao projeto de pesquisa), indica o maior grau de disposição do usuário a fornecer dados sensíveis.

4.2.1 Funcionamento e estrutura do subsistema servidor

O servidor elaborado neste experimento, foi o mesmo utilizado em todos os demais subsequentes. A principal atribuição deste sistema é fazer a a gerência do sítio que coletava dados de resposta aos estímulos enviados. Este sistema é o responsável por fazer as contabilizações descritas no início da seção 4.2.

Funcionamento

Para os usuários que acessarem o link contido no tuíte enviado, antes do direcionamento para uma página com informação legítima, lhes foi apresentada uma página do sítio da pesquisa mostrado na figura 4.7, que apresenta um formulário e solicita informação de dados pessoais do usuário, para um suposto cadastro. Esse acesso foi contabilizado. Na página foram apresentados dois botões: “Acessar” e “Acessar sem cadastro”, e também um link pouco evidente, intitulado “Para ver o projeto desta pesquisa científica clique aqui”, cujo link direciona o usuário à leitura do projeto da pesquisa aqui relatada. O eventual acesso do usuário ao projeto de pesquisa foi contabilizado.

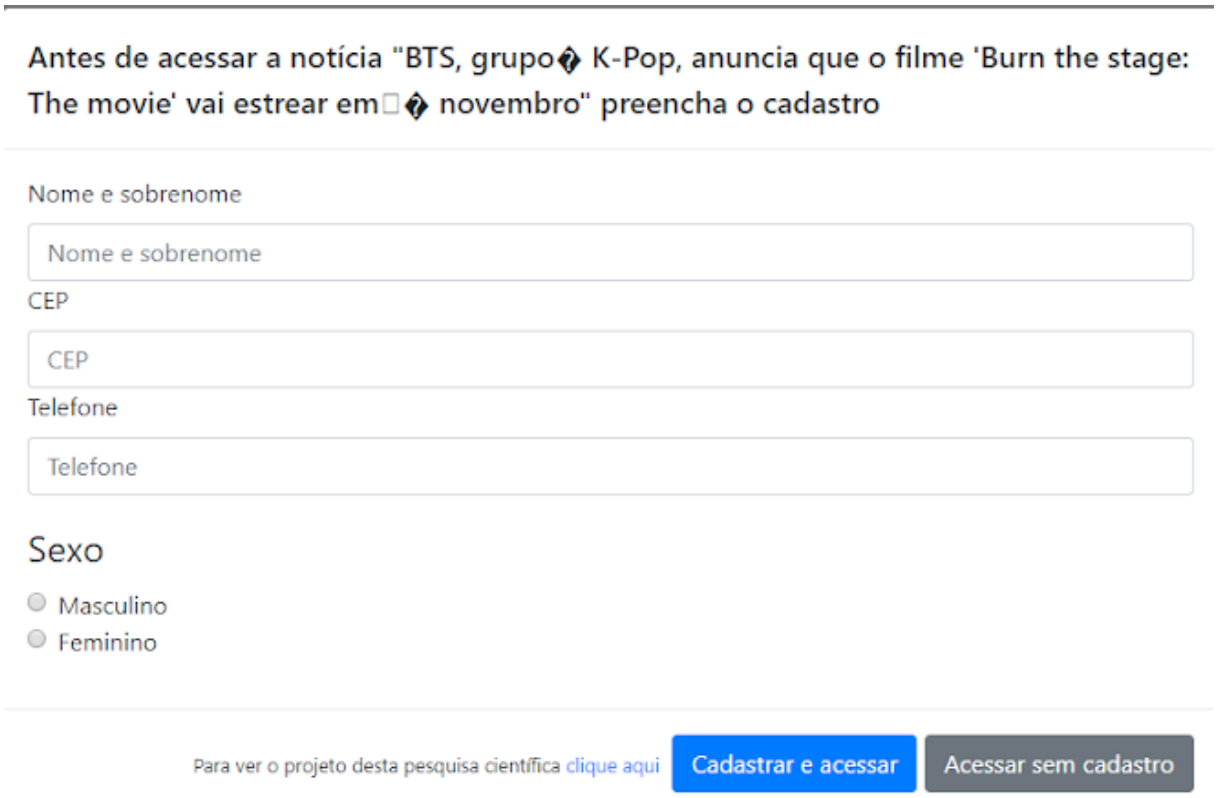

Figura 4.7: Formulário para coleta de dados durante pseudo-ataques de *phishing*..

Ao clicar no botão de “Acessar sem cadastro”, independentemente do preenchimento de dados do formulário, o usuário foi redirecionado para uma página legítima que continha informações verdadeiras, abordando o assunto relacionado aos termos informados no passo 1. Foi contabilizado o eventual acesso do usuário à notícia, sem a realização do cadastro.

O botão “Cadastrar e acessar” foi acionável apenas caso tenham sido preenchidos dados não nulos em todos os campos do formulário. Do mesmo modo que ocorre com o pressionamento do botão “Acessar sem cadastro”, a "vítima"foi direcionada para uma

página legítima que continha informações verdadeiras, abordando o assunto relacionado aos termos informados no passo 1. O eventual acesso do usuário à notícia, com a suposta realização do cadastro, foi contabilizado. Os dados informados pelo usuário não foram enviados ao servidor, nem armazenados de qualquer forma.

Estrutura

A figura 4.8 apresenta a arquitetura estática do experimento 2, isso é, sua estrutura. Podemos notar que o sistema apresenta uma estrutura quase monolítica, possuindo grande parte de sua lógica concentrada em um único módulo o módulo *Flask_app*.

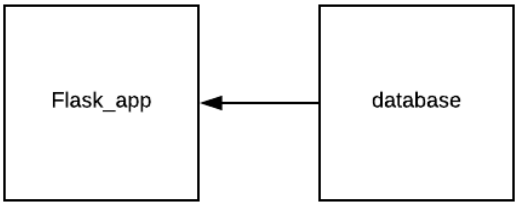

Figura 4.8: Diagrama que ilustra a organização dos módulos do servidor.

O módulo *Flask_app* foi responsável por renderizar a página web da pesquisa, e conforme chegavam as requisições *HTTP* ele contabilizava as interações que o usuário teve com a interface.

O módulo *database* tinha como função persistir as informações contabilizadas pelo módulo *Flask_app*. Para a persistência foi utilizado o SGBD *Mysql*. A persistência foi realizada em uma única tabela chamada 'AttackSimulation' a tabela possui os seguintes atributos:

- *id*: *int*(10), não nulo e chave primária;
- *url*: *varchar*(255), não nulo;
- *subject*: decimal(9,2), não nulo;
- *authority*: decimal(9,2), não nulo;
- *user_id*: *varchar*(255), não nulo;
- *entered_on*: *datetime*, não nulo;
- *viewed_project*: *datetime*;
- *without_fill*: *datetime*;

- *filled*: *datetime*;

4.2.2 Arquitetura/Desenvolvimento do software utilizado no experimento 2

Nesta seção serão discutidos aspectos que permearam o desenvolvimento do software utilizado no experimento 2. Esse foi composto por 2 subsistemas, um cliente e um servidor esse último já explicado na seção 4.2.1. O cliente executou a lógica dos robôs (bots), enquanto o servidor hospedou o sítio da coleta de dados.

Estrutura do subsistema cliente no experimento 2

A figura 4.9 apresenta a arquitetura estática do software usado no experimento 2, isso é, sua estrutura. Em um nível de abstração mais alto, o subsistema Cliente executa duas tarefas elementares distintas. Elas são executadas simultaneamente através de processos leves (threads), descritos após a apresentação dos módulos:

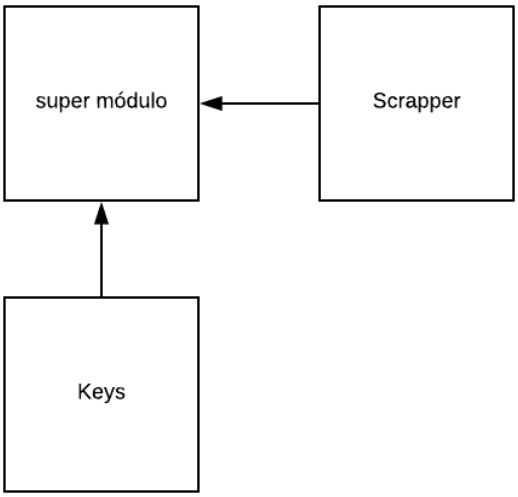

Figura 4.9: Diagrama que ilustra a organização dos módulos do cliente no experimento 2.

Podemos notar que o sistema apresenta uma estrutura quase monolítica, possuindo grande parte de sua lógica concentrada em um super módulo. Os módulos, detalhados a seguir, são:

1. *Keys*
2. *Scrapper*
3. Super módulo

Módulo Keys Devido à diversidade de contas disponíveis aos pesquisadores para a criação de robôs descrita na seção 2.5.4, o software foi desenvolvido de modo que mais de um robô pudesse estar executando ao mesmo tempo. Para isso, a cada conta que realiza uma conexão com a *API* do twitter é criado um par de *threads* que executam as tarefas supracitadas.

Para a realização das conexões com a *API* de maneira mais eficiente e segura, o módulo *keys* é utilizado para fazer a gerência das chaves, estas são armazenadas em um arquivo possuindo um formato específico e no momento que o *BOT* deseja conectar-se com a *API* ele solicita as chaves ao módulo, que as retorna.

Módulo Scrapper Conforme foi explicado na seção 1.3, um dos elementos de um ataque de *phishing* é uma isca que seja atraente o suficiente para que as vítimas se sintam tentadas a cair no golpe. As iscas aqui são notícias, reais obtidas diretamente, de forma automatizada, do portal de notícias G1. Na implementação do experimento 2, este módulo sempre usava a primeira notícia da página do editorial referente ao assunto do *BOT*.

Super módulo Esse suporte módulo agregava as demais funcionalidades necessárias para realizar o procedimento descrito no início da seção 4.2.

Funcionamento do experimento 2

Todos os módulos do cliente se combinam, buscando executar as threads *streaming* e seleção de alvos. O funcionamento destas esta descrito a seguir.

Thread Streaming O comportamento da *thread streaming* ocorre conforme a máquina de estados mostrada na figura 4.10. Aqui ocorre a captura dos tuítes realizados pelos usuários do twitter em tempo real. Sempre que há a ocorrência de uma das palavras-chave mostradas na seção 4.1.1 em algum tuíte, a API encaminha esse para a aplicação. Ao receber este, a aplicação realiza uma filtragem de acordo com alguns critérios estabelecidos pelos pesquisadores, buscando obter mais dados coesos e coerentes. As regras são as seguintes:

- Foram eliminados os tuítes cujo o idioma de escrita não é a língua portuguesa. Essa característica é importante porque o foco do estudo são os usuários brasileiros.
- São eliminados os tuítes que são provenientes de contas que não seguem nenhuma conta. Essa regra existe para eliminar contas que não são ativas na mídia social. assume-se que o *twitter* possui pouca utilidade para um usuário humano. Caso esse não siga nenhuma conta.

Se o tuíte não é aprovado pela filtragem, ele é ignorado. Caso contrário, um alvo é inserido na fila de alvos referente ao assunto do tuíte. Um alvo consiste em um objeto de dados com os seguintes atributos sobre a conta que realizou o tuíte: Autoridade, *username*, número de identificação fornecido pelo *twitter* e o assunto de interesse.

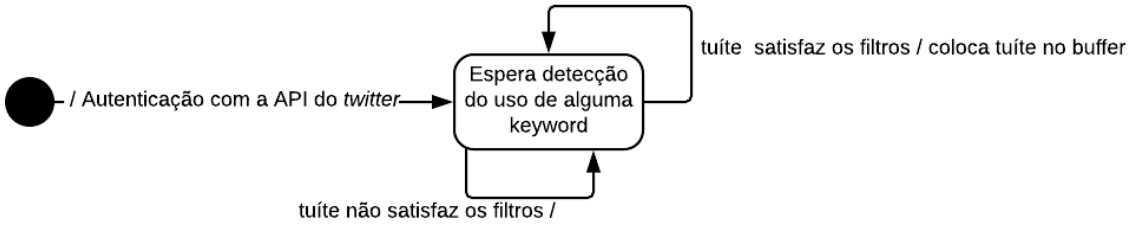

Figura 4.10: Máquina de estados que representa o comportamento da *thread Stream*.

Toda a lógica de *streaming* está inclusa no módulo batizado de '*Client*'. Este é o módulo principal do cliente, sendo responsável também pelas conexões com a *API* e gerência das *threads*.

Thread Seleção de alvos Esta é a segunda tarefa que o lado do cliente da aplicação realiza. Ela é intrinsecamente ligada à tarefa 1, pois para que o robô possa selecionar os alvos, e realizar os pseudo-ataques, ele precisa saber quais foram as contas que manifestaram interesse nos assuntos escolhidos, através do uso das *keywords*.

A autoridade calculada pelo *FollowerRank*(seção 2.6.3) é um número entre 0.0 e 1.0. Portanto, para este experimento são consideradas 10 faixas de autoridade possíveis. Os alvos foram escolhidos conforme a faixa da autoridade que a conta que escreveu o tuíte ocupa. No início, o robô procura uma conta que usou a palavra-chave, no seu assunto de interesse, iniciando com a faixa 0 - que são contas com a autoridade entre 0.00 e 0.09 -. Assim que um tuíte cuja conta possui autoridade nessa faixa de interesse é encontrado, é enviada a isca, e o robô continua procurando um alvo na próxima faixa. A descrição detalhada das faixas consideradas encontra-se na tabela 4.3.

O diagrama de estados da figura 4.11 descreve o funcionamento da *thread* de seleção de alvos para o experimento 2.

4.2.3 Execução do experimento 2

O experimento 2 foi executado nos dias 18 e 19 de outubro de 2018, em duas baterias sendo uma curta de 30 minutos no dia 18 e outra longa de 3 horas no dia 19 (até o banimento das contas). A codificação deste foi realizada no computador 2, e sua execução foi feita

Id da faixa de autoridade	limite da faixa
0	de 0,00 a 0,09
1	de 0,10 a 0,19
2	de 0,20 a 0,29
3	de 0,30 a 0,39
4	de 0,40 a 0,49
5	de 0,50 a 0,59
6	de 0,60 a 0,69
7	de 0,70 a 0,79
8	de 0,80 a 0,89
9	de 0,90 a 0,99

Tabela 4.3: Descrição detalhada das faixas de autoridade adotadas

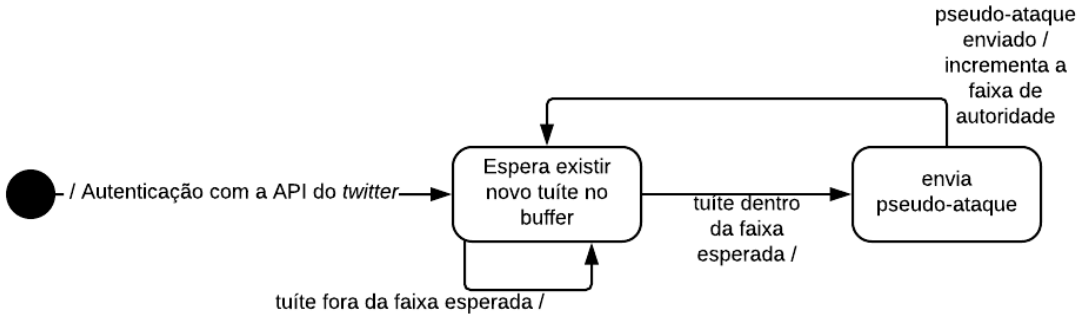

Figura 4.11: Máquina de estados que representa o comportamento da *thread* de seleção de alvos no experimento 2.

usando o computador 1. A descrição completa das máquinas utilizadas encontra-se na seção 3.5.

A duração dos pseudo-ataques durou apenas algumas algumas horas, devido a um bloqueio automático imposto pelo *twitter*. Utilizaram-se 2 contas diferentes(conta4 e conta5), onde cada conta realizava postagens específicas em uma das áreas temáticas: esportes ou entretenimento. Foram enviados 65 tuítes (32 em uma conta e 33 em outra) para usuários distintos, e esses foram segmentados em 10 faixas de autoridades. Os pesquisadores optaram por não realizar testes com o assunto política, pois esses foram realizados durante o período da campanha eleitoral brasileira no ano de 2018, onde havia em jogo vagas para cargos de presidente, senador, governador, deputado federal e deputado estadual.

Como já dito, essa primeira bateria de testes foi interrompida devido a um bloqueio realizado pelo *twitter* em ambas as contas, onde estas foram proibidas de escrever novos tuítes através da *API*. Entretanto, apesar do pouquíssimo tempo de execução, foi possível

obter alguns resultados interessantes.

4.2.4 Dados produzidos do experimento 2

Durante 3 dias de execução dos pseudo-ataques do experimento 2, os 65 links enviados estimularam 51 acessos. A maioria destes foi provenientes de notícias sobre esportes, totalizando 50 acessos, enquanto apenas 1 acesso foi realizado através de uma notícia referente a entretenimento.

Os 50 acessos ao servidor, realizados em decorrência de 10 tuítes (pseudo-ataques) enviados com referências aos títulos das notícias sobre esportes. Esses 10 tuítes foram enviados para 10 contas distintas, que possuíam autoridade (calculada através do *followerRank*) bem distribuída nas 10 faixas. O acesso realizado no link sobre entretenimento, foi realizado por uma conta pertencente a faixa de autoridade 4, ou seja entre 0,40 e 0,49. Além disso alguns tuítes da conta 'conta5' conseguiram atrair algum engajamento de outras contas do *twitter*, atraindo alguns retuítes e curtidas conforme mostrado na Figura 4.13.

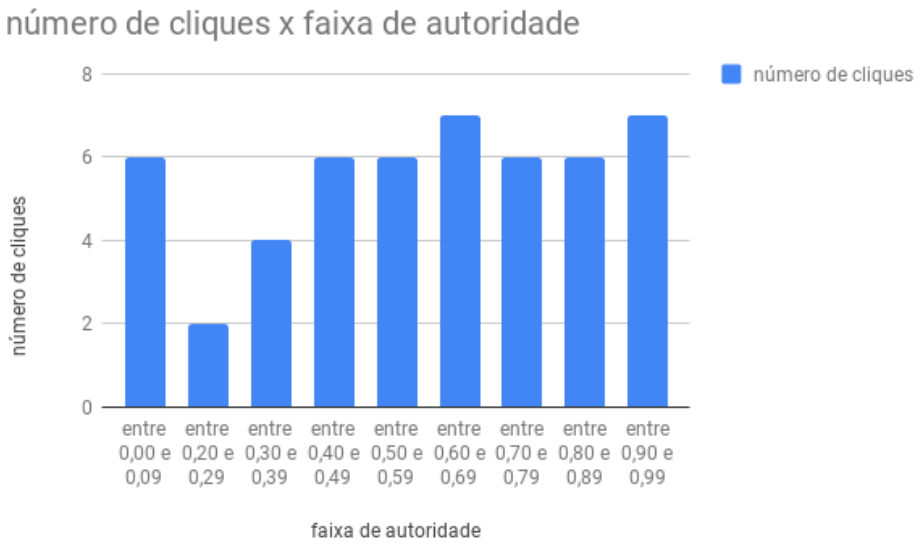

Figura 4.12: Distribuição de cliques por faixa de autoridade no assunto esportes.

O gráfico da figura 4.12 sugere que as autoridades maiores que 0,5 tendem a gerar mais interação com a isca, entretanto, os dados coletados por ora não podem ser usados para gerar qualquer afirmação mais impactante.

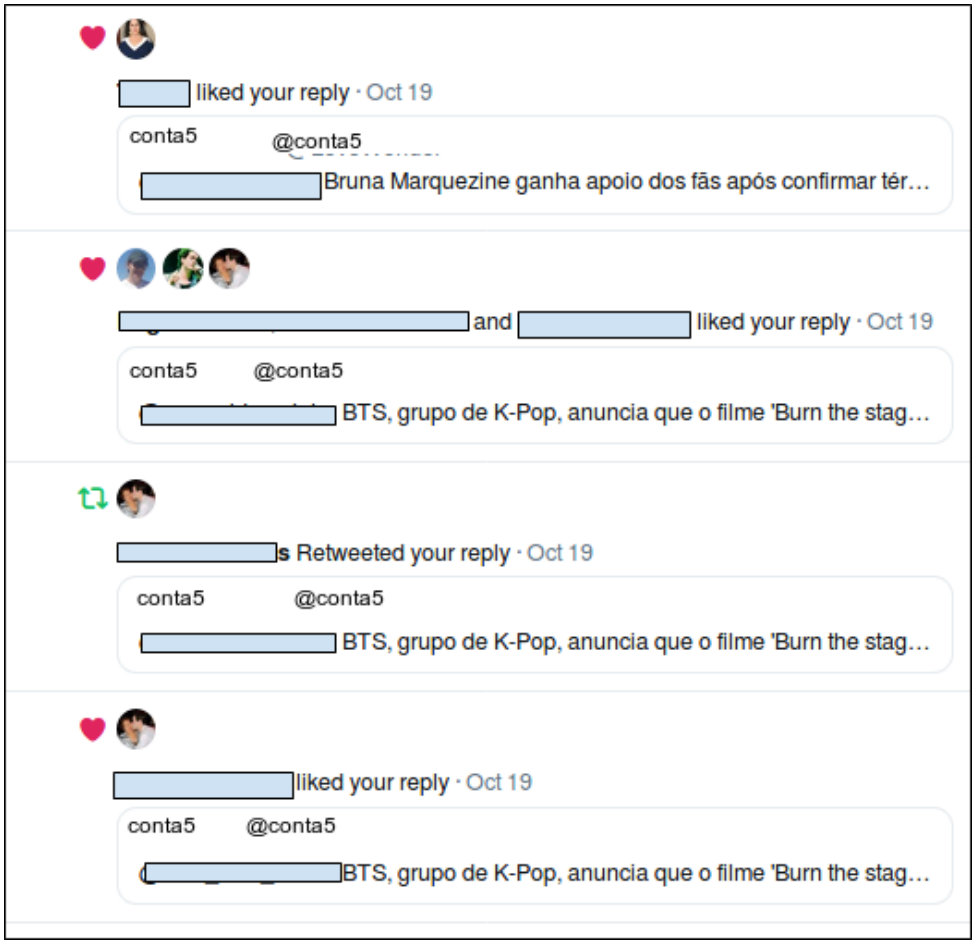

Figura 4.13: Retuítes e curtidas obtidos pela conta conta5.

4.2.5 Aprendizagem gerada com o experimento 2

Após a realização do experimento 2 foi compreendido que o programa de computador desenvolvido não atendia aos requisitos funcionais estabelecidos. O *BOT* deve ser bem mais sutil para que possa passar desapercebido pelos mecanismos de detecção do twitter. Além disso, notou-se uma necessidade de maior modularização do código-fonte, objetivando uma maior manutenibilidade do software.

Esse experimento fomentou um refinamento nos requisitos dos *BOTs*. Através de uma nova leitura mais detalhada das regras de uso da *API* do twitter [] perceberam-se os seguintes pontos em que os robôs estavam falhando:

1. O robô não realizava postagens sem fazer menções a outras contas;
2. O robô realizava muitas postagens duplicadas, ou muito semelhantes;
3. O robô realizava o redirecionamento dos usuários para uma página intermediária antes de enviar para a notícia.

4.3 Experimento 3 - Tornando mais clara a estratégia para a realização de de pseudo-ataques

O objetivo do experimento 3 foi refinar a estratégia de testes com *phishing* criada no experimento 2. As melhorias implementadas foram baseadas no aprendizado obtido com o experimento anterior. Na seção 4.2.5 foram citados alguns pontos que, segundo as regras oficiais de uso da *API* do *twitter*, o robô do experimento 2 feria. Para cada regra identificada foi elaborada uma estratégia de mitigação, tentando fazer com que os BOTs passasem despercebidos. Essas estratégias são as seguintes:

O robô realizava muitas menções a outras contas A estratégia de mitigação para esse caso foi fazer com que o robô realizasse a postagem de alguns outros tuítes intercalados com os tuítes de pseudo-ataques. Para isso foi incorporado o módulo `phrase generator`, detalhado na seção 4.3.1.

O robô realizava muitas postagens duplicadas ou muito semelhantes Foi adotada uma estratégia de diversificação das iscas enviadas. Essa diversificação causou mudanças no módulo `Scrapper` e no módulo `Messenger`. O módulo `phrase generator` já posta tuítes que não são semelhantes, dispensando preocupação quanto a esse tópico. Os módulos são detalhados na seção 4.3.1.

O robô realizava o redirecionamento para uma página intermediária Esse problema não foi totalmente resolvido, mas as iscas do experimento 2 passavam às claras a *URL* para a qual a página intermediária iria fazer o redirecionamento. Esse problema foi resolvido através da adoção do módulo `short`,detalhado na seção 4.3.1.

Após a implementação, iniciaram-se os ensaios com usuários reais. Esses ocorreram durante 2 dias ininterruptos. Os testes foram interrompidos pelos pesquisadores, devido a um bloqueio automático imposto pelo *twitter* a uma das contas utilizadas. Foram usadas 3 contas diferentes (conta1, conta2 e conta3), onde cada conta realizava postagens específicas em uma das áreas temáticas: política, esportes ou entretenimento. A conta3 foi bloqueada para uso da *API*.

O experimento 3 foi elaborado baseando-se no experimento 2, e por isso possui a sequencia de passos muito semelhante. A única diferença encontra-se na inclusão de um passo adicional, onde o robô realiza uma ação quando o provável alvo não é selecionado. O experimento foi estruturado nos seguintes passos:

1. Foi obtido do *twitter* um fluxo contínuo de tuítes associados a um conjunto de termos que representam assuntos em política, esportes ou entretenimento;

2. Na medida em que o fluxo de tuítes foi produzido, realizou-se uma amostragem das contas, com base na medida quantitativa da autoridade dessas contas;

3. Para as contas da amostra, foram enviados novos tuítes, oferecendo informações sobre temas de interesse associados ao assunto investigado, contendo um link que oferece informações legítimas sobre o assunto investigado;

4. Para cada conta detectada que não entrou na amostra, o programa de computador realizava uma das seguintes ações: (1) em 75% das vezes a ação tomada era a realização de uma postagem referente ao assunto que o BOT procurava, mas sem realizar nenhuma menção ou possuir um link na postagem. (2) em 25% das vezes a ação efetivada era seguir a conta que não entrou na amostragem;

5. Para os usuários que acessarem o link contido no tuíte enviado, antes do direcionamento para uma página com informação legítima, lhes foi apresentada uma página do sítio da pesquisa, que apresenta um formulário e solicita informação de dados pessoais do usuário, para um suposto cadastro. Esse acesso foi contabilizado. Na página foram apresentados dois botões: "Acessar" e "Acessar sem cadastro", e também um link pouco evidente, intitulado "Para ver o projeto desta pesquisa científica clique aqui", cujo link direciona o usuário à leitura do projeto da pesquisa aqui relatada. O eventual acesso do usuário ao projeto de pesquisa foi contabilizado.

6. Ao clicar no botão de "Acessar sem cadastro", independentemente do preenchimento de dados do formulário, o usuário foi redirecionado para uma página legítima que contém informações verdadeiras, abordando o assunto relacionado aos termos informados no passo 1. Foi contabilizado o eventual acesso do usuário à notícia, sem a realização do cadastro.

7. O botão "Cadastrar e acessar" foi acionável apenas caso tenham sido preenchidos dados não nulos em todos os campos do formulário. Do mesmo modo que ocorre com o pressionamento do botão "Acessar sem cadastro", o usuário foi redirecionado para uma página legítima que contém informações verdadeiras, abordando o assunto relacionado aos termos informados no passo 1. O eventual acesso do usuário à notícia, com a suposta realização do cadastro, foi contabilizado. Os dados informados pelo usuário não foram enviados ao servidor, nem armazenados de qualquer forma.

8. foram contabilizadas:(1) as contas amostradas segundo o uso das palavras-chave, para as quais foram enviados pseudo-ataques; (2) quais as autoridades dessas contas, obtidas por uma relação entre a quantidade de contas de seguidores e a quantidade de contas seguidas no *twitter*; (3) se estas contas acessaram o link enviado no pseudo-ataque; (4) Se o usuário da conta pseudo-atacada acessou o projeto da pesquisa; (5)

se o usuário da conta pseudo-atacada acionou o botão “Cadastrar e acessar”, o que ocorre apenas se inseriu uma quantidade de caracteres não vazia nos campos que supostamente coletaram seus dados pessoais; (6) Se o usuário da conta pseudo-atacada acionou o botão “Acessar sem cadastro”. Nos casos 5 e 6, considera-se que foi apresentada a página que remete à informação de interesse do usuário. O caso 5 (preenchimento do formulário), sem que tenha havido o caso 4 (acesso ao projeto de pesquisa), indica o maior grau de disposição do usuário a fornecer dados sensíveis.

4.3.1 Arquitetura/Desenvolvimento do experimento 3

Nessa seção serão discutidos aspectos que permearam o desenvolvimento do software utilizado no experimento 3. Esse foi composto por 2 subsistemas, um cliente e um servidor. O cliente executou a lógica dos robôs, enquanto o servidor hospedou o sítio da pesquisa. Conforme já explicado na seção 4.2.1, o servidor utilizado foi o mesmo durante todos os experimentos.

Estrutura do subsistema cliente no experimento 3

A figura 4.14 apresenta a arquitetura estática do experimento 3, isso é, sua estrutura. Em um nível de abstração mais alto, o subsistema Cliente executa duas tarefas elementares distintas. Elas são executadas simultaneamente através de processos leves (*threads*), descritos após a apresentação dos módulos.

Podemos notar que o sistema foi refatorado, e comparando ao experimento 2, ele possui um número muito maior de componentes. Os módulos, detalhados a seguir, são:

1. *Keys*
2. *Scrapper*
3. *Phrase generator*
4. *Short*
5. *Client*
6. *Messenger*

Módulo Keys Este módulo é o mesmo utilizado no experimento 2.

Módulo Scrapper Este faz um *web scrapping* em tempo real dos editoriais de notícias sobre política, esportes(Globo Esporte) e entretenimento do portal G1, para que as noticias usadas como isca sejam sempre atuais e potencialmente mais atraentes.

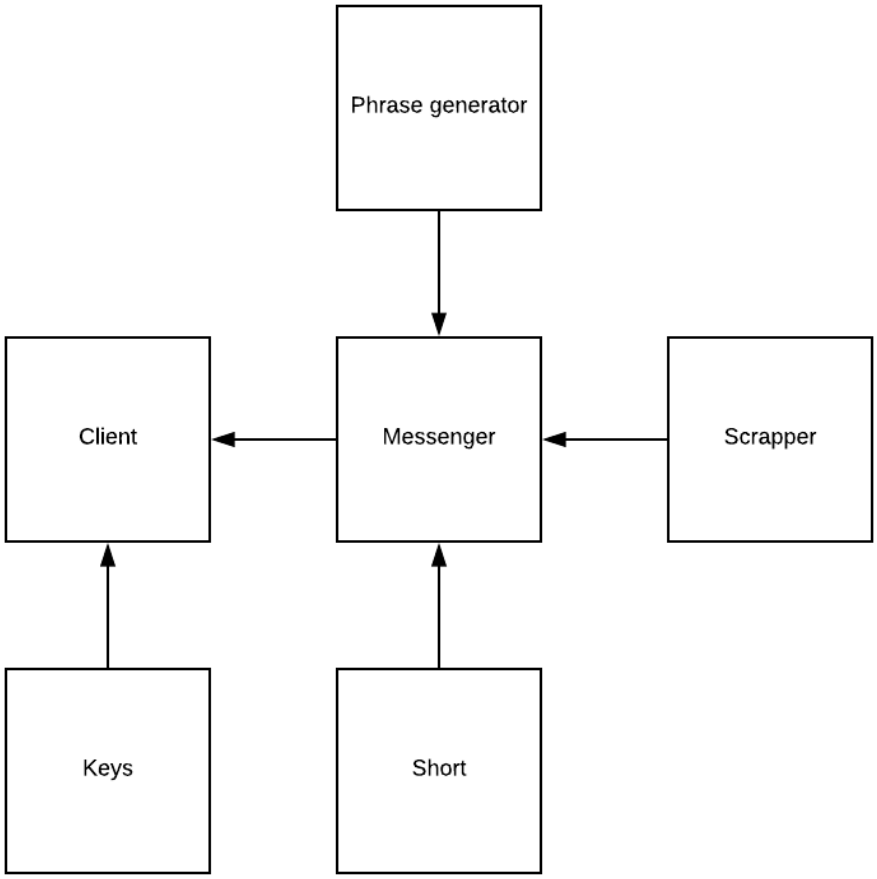

Figura 4.14: Organização dos módulos do cliente no experimento 3.

Para isso, sempre que um alvo é identificado, o módulo *Scrapper* se direciona ao editorial de interesse desse alvo e, diferentemente do procedimento realizado no experimento 2, escolhe uma notícia de maneira pseudo-aleatória, dentre as disponíveis no portal acessado. Essa escolha suficientemente randômica é importante para gerar uma maior variabilidade de *links* a serem enviados, dificultando que os robôs sejam identificados como usuários não humanos.

Módulo *Phrase generator* Uma das maiores dificuldades em criar um *BOT* sem dúvidas é conseguir camuflá-lo dentro da plataforma. Devido ao grande número de robôs que atuam na rede mundial de computadores, o *twitter* dá uma atenção grande ao uso de robôs, principalmente aos que possuem um comportamento aparentemente malicioso, como é o caso deste trabalho.

Para mascarar as atividades dos robôs da pesquisa, esses possuíam a habilidade de realizar tuítes assemelhando-se a usuários humanos. Para realizar essa tarefa, utilizou-se

a biblioteca *opensource* Markovbot que gera frases a partir de um conjunto de textos inseridos, tal como o texto de um livro.

O módulo *Phrase generator* serve como interface ao *Markovbot*. Nesse módulo são instanciados 3 *Markovbots*, onde cada um gerava frases a partir do texto de um livro sobre um dos assuntos escolhidos para a pesquisa (seção 4.1.1). Dessa forma sempre que um robô do experimento solicita, é criada um tuíte que é relacionado ao assunto que *BOT* escreve.

Os livros que os BOTs se baseiam para escrever os tuítes (figura 4.15) são relacionados as áreas temáticas. Para a área política, o livro utilizado é 'O ódio como política: a reinvenção das direitas no Brasil' da autora Esther Solano [42], para esportes o escolhido foi 'Neymar: o sonho brasileiro' do autor Peter Banker [43]. Para entretenimento utilizou-se 'O livro da astrologia: Um guia para céticos, curiosos e indecisos' do autor Carlos Orsi [44]. Os livros foram escolhidos com base na afinidade com as *hashtags* que definiram os tuítes ligados ao tema, combinada com disponibilidade de acesso ao texto em formato online, por meio de buscas no *Google* e outros sítios. Os conteúdos dos livros não foram estudados pelos pesquisadores.

Figura 4.15: Capas dos livros utilizados pelos robôs para gerar tuítes.

Módulo Short Este módulo é bastante simples, e consiste em uma interface com o encurtador de links *TinyURL*. Além de poupar caracteres, um recurso importante devido ao tamanho máximo que um tuíte pode ter, esse módulo também foi importante para camuflar a *URL* da notícia que é acessada após a passagem pelo sítio da pesquisa.

Módulo *Client* O módulo *client*, como o nome sugere, é o módulo central do lado cliente da aplicação. Nele os outros módulos se ligam para realizar todas as tarefas neces-

sárias. A primeira funcionalidade que esse módulo realiza é a gerência da(as) conexão(ões) com a *API* do twitter, com o auxílio das chaves fornecidas pelo módulo *keys*.

Com a(as) conexão(ões) estabelecida(as), são criadas 2 *threads* para cada conexão. Uma *thread* é responsável pela tarefa de *streaming* e a outra pela tarefa de seleção de alvos. A lógica da tarefa de *streaming* está inteiramente contida nesse módulo, enquanto a lógica da seleção de alvos está no módulo messenger. Essas tarefas são melhor descritas nas seção 4.2.2. Resumidamente o módulo é responsável por fazer a gerência das *threads*, consequentemente gerência das conexões também, e de realizar as interações com a API, que recebem os dados.

Módulo Messenger Ao contrário do modulo *client*, esse é o responsável por fazer requisições para a *API tweepy*, que envia dados para a *API*. Conforme descrito na figura 4.14, o módulo *Messenger* faz interface com diversos outros módulos, a maioria deles vislumbrando cumprir a tarefa de seleção de alvos (seção 4.2.2). Entretanto, ele é o responsável também pela postagem de tuítes que não são pseudo-ataques, utilizando os tuítes gerados pelo módulo *Phrase Generator*.

Funcionamento da aplicação do experimento 3

O funcionamento da aplicação do experimento 3 é em grande parte semelhante ao funcionamento do experimento 2. A única mudança ocorreu na *thread* seleção de alvos.

A tarefa de *streaming* ocorre conforme o diagrama de estados da figura 4.10. A mudança ocorrida foi na *thread* seleção de alvos. Essa é relacionada apenas aos alvos que não foram selecionados. No experimento 2 quando um potencial alvo não era selecionado, o *BOT* não realizava nenhuma ação.

Durante o experimento 3, sempre que o potencial alvo é retirado do *buffer* não é escolhido, o software executa uma das seguintes ações de maneira pseudoaleatória:

- Em 75% das vezes ele realiza a postagem de um tuíte com o conteúdo referente a área temática do *BOT*. Neste tuíte não há nenhum link ou menção a outra conta no twitter;
- Em 25% das vezes ele se registra como seguidor da a conta que foi retirada do *buffer*.

Estas ações são importantes para tornar o comportamento dos BOTs mais semelhante ao comportamento de um usuário humano. Tornando os robôs mais parecidos com humanos, é possível camuflar melhor a existência deles na mídia social, tornando-os mais longevos, quando comparados aos robôs do experimento 2. O diagrama de estados da figura 4.16 ilustra de maneira precisa o funcionamento da *thread* de seleção de alvos para o experimento 3.

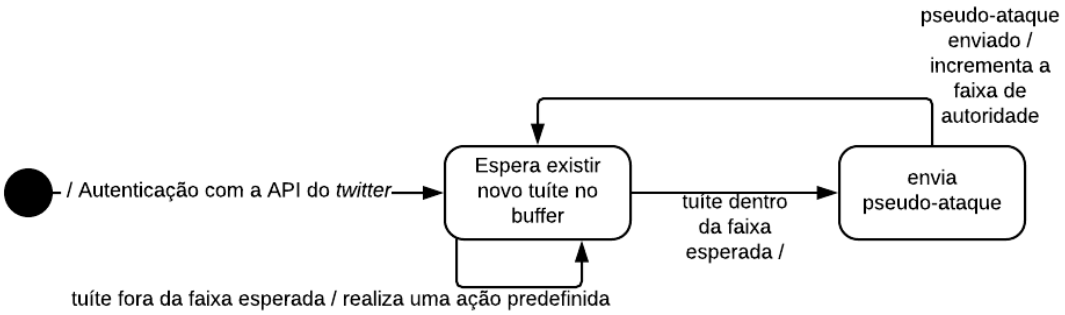

Figura 4.16: Máquina de estados que representa o comportamento da *thread* de seleção de alvos no experimento 3.

Política	Esportes	Entretenimento	**TOTAL**
336	160	245	**741**

Tabela 4.4: Envios de ataques por assunto no experimento 3.

4.3.2 Execução do experimento 3

O experimento 3 foi executado entre os dias 12 e 15 de novembro de 2018, até o banimento de uma das contas. A codificação do experimento foi realizada no computador 2, e sua execução foi feita usando o computador 1 seção 3.5.

4.3.3 Dados produzidos com o experimento 3

O experimento 3 rodou durante pouquíssimo tempo, e consequentemente não produziu nenhum dado que por si só fosse relevante. Igual ao ocorrido no experimento da seção 4.2. Por isso os dados de acessos referentes a este período foram incorporados aos dados do experimento 4 (seção 4.4) pois não há mudanças metodológicas que prejudicam a análise destes dados de maneira conjunta.

Entretanto existiram fortes indicativos de que o desenvolvimento dos experimentos estava progredindo de maneira satisfatória. Apesar do banimento de uma das contas, essa conta ficou muito mais tempo executando, se comparado ao experimento 2 (seção 4.2). A tabela 4.4 mostra a quantidade de ataques enviados no experimento 3, segmentados por área temática.

4.3.4 Aprendizagem técnica com o experimento 3

O banimento de um dos robôs (conta1) durante a realização do experimento 3 ocasionou um estado de alerta nos pesquisadores. Não se sabe ao certo o porquê da conta ter sido

banida. Há três razões possíveis para explicar o que poderia ter ocasionado o banimento. A primeira é de que algum dos alvos fez uma denúncia ao *twitter*. A segunda seria de que o *BOT* teve atividade 24h, sem nenhuma pausa. A terceira é que o *BOT* realizava postagens com muita frequência.

4.4 Experimento 4 - Experimento final

O experimento 4 foi o último realizado na pesquisa e visava conseguir um arcabouço definitivo para desenvolver testes com pseudo ataques de *phishing*. Semelhante ao ocorrido no experimento 3, os aprendizados obtidos no experimento anterior serviram para embasar as melhorias na nova versão do programa de computador. Desse modo foram adotadas medidas de mitigação para as 3 razões possíveis do banimento da conta conta1.

Caso o BOT tenha sofrido uma denúncia, não há o que fazer para evitar que esse tipo de coisa aconteça. Se a razão do banimento foi uma das outras , haveriam de ser adotadas algumas medidas. Percebeu-se que os experimentos não poderiam ser executados de maneira ininterrupta. Por isso, durante a madrugada, os experimentos eram interrompidos. Além disso, constatou-se que os robôs deveriam atuar de maneira mais lenta, para que esses ficassem mais parecidos com o comportamento humano. A diminuição da velocidade de postagem do tuítes ocorreu através de uma solução bastante simples: desprezando uma parte dos alvos em potencial.

Após definidas as estratégias, como o objetivo de evitar o banimento das contas restantes, iniciou-se a nova bateria de testes. O experimento 4 foi elaborado baseando-se no experimento 3, e por isso possui a sequência de passos muito semelhante. O experimento foi estruturado nos seguintes passos:

1. Foi obtido do *twitter* um fluxo contínuo de tuítes associados a um conjunto de termos que representam assuntos em política, esportes ou entretenimento;
2. Na medida em que o fluxo de tuítes foi produzido, realizou-se uma amostragem das contas, com base na medida quantitativa da autoridade dessas contas. Entretanto para diminuir a velocidade na qual as postagens era realizada desprezou-se 2/3 (dois terços) do fluxo de tuítes original antes de realizar a amostragem;
3. Para as contas da amostra, foram enviados novos tuítes, oferecendo informações sobre temas de interesse associados ao assunto investigado, contendo um link que oferece informações legítimas sobre o assunto investigado;
4. Para cada conta detectada que não entrou na amostra, o programa de computador realizava uma das seguintes ações: (1) em 75% das vezes a ação tomada era a

realização de uma postagem referente ao assunto que o BOT procurava, mas sem realizar nenhuma menção ou possuir um link na postagem. (2) em 25% das vezes a ação efetivada era seguir a conta que não entrou na amostragem;

5. Para os usuários que acessarem o link contido no tuíte enviado, antes do direcionamento para uma página com informação legítima, lhes foi apresentada uma página do sítio da pesquisa, que apresenta um formulário e solicita informação de dados pessoais do usuário, para um suposto cadastro. Esse acesso foi contabilizado. Na página foram apresentados dois botões: "Acessar" e "Acessar sem cadastro", e também um link pouco evidente, intitulado "Para ver o projeto desta pesquisa científica clique aqui", cujo link direciona o usuário à leitura do projeto da pesquisa aqui relatada. O eventual acesso do usuário ao projeto de pesquisa foi contabilizado.

6. Ao clicar no botão de "Acessar sem cadastro", independentemente do preenchimento de dados do formulário, o usuário foi redirecionado para uma página legítima que contém informações verdadeiras, abordando o assunto relacionado aos termos informados no passo 1. Foi contabilizado o eventual acesso do usuário à notícia, sem a realização do cadastro.

7. O botão "Cadastrar e acessar" foi acionável apenas caso tenham sido preenchidos dados não nulos em todos os campos do formulário. Do mesmo modo que ocorre com o pressionamento do botão "Acessar sem cadastro", o usuário foi redirecionado para uma página legítima que contém informações verdadeiras, abordando o assunto relacionado aos termos informados no passo 1. O eventual acesso do usuário à notícia, com a suposta realização do cadastro, foi contabilizado. Os dados informados pelo usuário não foram enviados ao servidor, nem armazenados de qualquer forma.

8. foram contabilizadas:(1) as contas amostradas segundo o uso das palavras-chave, para as quais foram enviados pseudo-ataques; (2) quais as autoridades dessas contas, obtidas por uma relação entre a quantidade de contas de seguidores e a quantidade de contas seguidas no *twitter*; (3) se estas contas acessaram o link enviado no pseudo-ataque; (4) Se o usuário da conta pseudo-atacada acessou o projeto da pesquisa; (5) se o usuário da conta pseudo-atacada acionou o botão "Cadastrar e acessar", o que ocorre apenas se inseriu uma quantidade de caracteres não vazia nos campos que supostamente coletaram seus dados pessoais; (6) Se o usuário da conta pseudo-atacada acionou o botão "Acessar sem cadastro". Nos casos 5 e 6, considera-se que foi apresentada a página que remete à informação de interesse do usuário. O caso 5 (preenchimento do formulário), sem que tenha havido o caso 4 (acesso ao projeto de pesquisa), indica o maior grau de disposição do usuário a fornecer dados sensíveis.

Os testes foram interrompidos por decisão dos pesquisadores, não tendo ocorrido nenhum problema com a plataforma diferentemente dos experimentos 2 e 3. Utilizou-se 2 contas diferentes (conta1 e conta2), onde cada *BOT* realizava postagens específicas em uma das áreas temáticas: política ou esportes.

4.4.1 Aplicação do experimento 4

Nesta seção é discutido o desenvolvimento do software utilizado no experimento 4. Esse foi composto por 2 subsistemas, um cliente e um servidor. O cliente executou a lógica dos robôs, enquanto o servidor hospedou o sítio da pesquisa. Conforme já explicado na seção 4.2.1, o servidor utilizado foi o mesmo durante todos os experimentos. Para esse experimento não houveram mudanças arquiteturais em relação ao experimento anterior, os módulos continuam os mesmos e as funcionalidades também.

Funcionamento da aplicação do experimento 4

O funcionamento da aplicação do experimento 4 é em grande parte semelhante ao funcionamento no experimento 3 descrito na seção 4.2.2. A única mudança ocorreu na *thread* seleção de alvos.

A tarefa de *streaming* ocorre conforme o diagrama de estados da figura 4.10. A mudança ocorrida foi na *thread* seleção de alvos. Agora, antes que os alvos passem por qualquer tipo de análise, dois terços desses é imediatamente desprezado de maneira pseudoaleatória.

Realizando essa estratégia, há uma diminuição considerável das ações do robô tornando essas mais paulatinas. Isso ocorre porque a realização de todas as ações dos *BOTs* acontece de acordo com a seleção de alvos (conforme descrito na seção 4.3.1). O diagrama de estados da figura 4.17 ilustra de maneira precisa o funcionamento da thread de seleção de alvos para o experimento 4.

4.4.2 Execução do experimento 4

O experimento 4 foi executado entre os dias 16 e 25 de novembro de 2018, até a decisão dos pesquisadores de parar com a experimentação. A codificação do experimento foi realizada no computador 2, e sua execução foi feita usando o computador 1, seção 3.5.

4.4.3 Dados produzidos no experimento 4

Durante toda a execução do experimento 4 foram enviados 569 links, distribuídos entre os assuntos política e esportes. Se comparado com o experimento anterior (tabela 4.4)

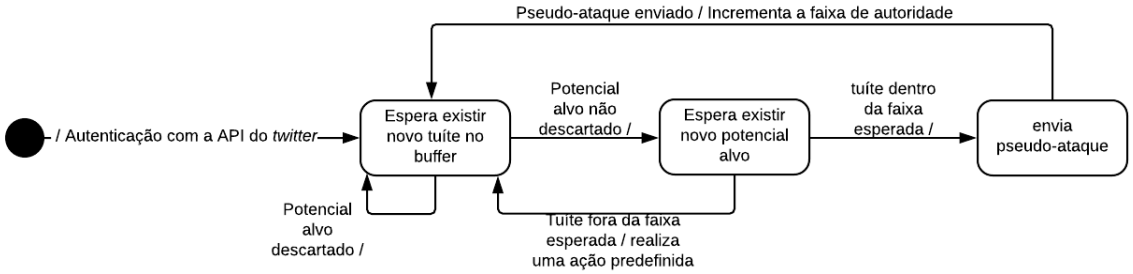

Figura 4.17: Máquina de estados que representa o comportamento da *thread* de seleção de alvos no experimento 4.

Política	Esportes	**TOTAL**
353	216	**569**

Tabela 4.5: Envios de ataques por assunto no experimento 4.

foram enviados 17 links a mais no assunto política, e 56 links a mais no assunto esportes. A tabela 4.5 mostra a quantidade de ataques enviados no experimento 4 segmentados por área temática.

4.4.4 Aprendizagem técnica com o experimento 4

O experimento 4 serviu para mostrar evidências de que o software proposto, e concebido, funciona e é aplicável para ataques de engenharia social automatizada. Os robôs atuaram durante 10 dias sem ser detectados pelo twitter e conseguiram inclusive se passar por usuários humanos, o que é evidenciável pelas interações que aconteceram com os *BOTs*. Ficou claro que o armazenamento de alguns outros dados durante o processo de expe-

Faixa	Política	Esportes	Entretenimento*	Geral
0	74	46	0	120
1	65	36	0	101
2	76	41	2	119
3	65	33	0	98
4	58	42	0	100
5	72	44	0	116
6	73	32	0	105
7	68	35	2	105
8	80	32	0	112
9	52	24	3	79
TOTAL	**683**	**365**	**7**	**1055**

Tabela 4.6: Acessos ao sítio da pesquisa por faixa de autoridade no experimento 4.

rimentação também poderia ter contribuído com a análise de dados de modo a torná-la mais rápida e diminuindo seus custos.

A estratégia de persistir os acessos utilizando um sistema gerenciador de bancos de dados com os acessos foi ruim. Isto ocorreu porque os acessos aos links através de *BOTs* não foi considerada pelos pesquisadores durante a concepção do experimento. Por isso, nos acessos registrados através do SGBD existiam registros referentes a acessos que não eram provenientes de usuários humanos e por isso eram inúteis para os propósitos desta pesquisa. Devido à modelagem do banco de dados não é possível diferenciar quais os acessos legítimos dos acessos criados por robôs. Por isso, os dados armazenados no SGBD não foram utilizados.

Além disso, houve um aprendizado referente ao cruzamento dos dados. O uso de softwares de planilhas eletrônicas não é o ideal ao lidar com uma massa considerável de dados. Entretanto, medidas devem ser adotadas para diminuir a necessidade de cruzamento de dados.

4.5 Conclusão Parcial

Este é um dos capítulos fundamentais deste trabalho, pois através dos dados fornecidos aqui é possível conduzir uma análise que pode ajudar a explicar os fenômenos do mundo real investigados pelos pesquisadores. Segue o capítulo referente a Análise e discussão dos dados aqui apresentados. O próximo capítulo vislumbra responder as perguntas de pesquisa apresentadas no capítulo 1.

Capítulo 5

Análise e Discussão

O capítulo 5 analisa e discute os resultados obtidos com base no referencial teórico-conceitual apresentado no capítulo 2. Primeiramente é conduzida uma análise dos dados apresentados no capítulo 4, buscando confirmar se estes são capazes de explicar o fenômeno a ser estudado. Após isto é apresentada uma discussão da análise exposta, explicitando quais conclusões podem ser tiradas dos dados. Tudo isso possuindo como guia o referencial teórico-conceitual.

5.1 Preparação dos dados

Para que os dados extraídos pudessem ser utilizados foi necessário um trabalho de preparação destes. Este trabalho ocorreu segundo o que está descrito nas seções subsequentes.

5.1.1 Problemas com o módulo *database*

O módulo *dabatase*, usado para registro dos ataques e resposta aos estímulos, do servidor de coleta de dados, não coletou dados suficientes para a realização de analises. foi então necessário usar os logos do servidor web, para geração dos dados, de modo que os dados do *database* foram completamente descartados. A identificação desse problema se iniciou porque o tanto de acessos no *database* era de uma ordem de grandeza superior aos número de pseudoataques enviados pelos robôs. Isso pôde ser constatado através da comparação com a quantidade de acessos disponibilizados nos *logs* de acesso disponibilizados pelo *pythonanywhere*.

Ao estudar melhor os *logs*, perceberam-se alguns acessos que poderiam vir a prejudicar os dados coletados. Esses eram acessos provenientes de outros robôs. Isso foi constatado pois haviam assinaturas nos *logs* que identificavam estes acessos. Por exemplo, na figura 5.1 temos alguns registros presentes nos *logs* de acesso ao módulo servidor, e no log

em destaque (com a cor laranja) é notável a assinatura referente à um robô chamado LivelapBot.

Figura 5.1: Exemplo dos *logs* de acesso.

Além dos robôs, o serviço utilizado para encurtar a *url* também realizava um acesso ao sítio da pesquisa por cada *link* gerado. Entretanto, só foi constatado esse problema o dia 29 de novembro, e o *pythonanywehre* só armazena 12 dias de *logs* de acesso, de modo que os dados disponíveis para analise foram apenas aqueles entre o dia 17 de novembro e o dia 29 de novembro. ou seja, dispúnhamos de 12 dias de dados referentes ao funcionamento dos robôs, para realização de analises.

Objetivando tornar os dados provenientes dos *logs* utilizáveis, foi realizada uma limpeza. Essa limpeza foi realizada em alguns passos:

1. Todos os arquivos de *logs* de acesso foram juntos em um único arquivo
2. Foram excluídos os registros que eram provenientes do *Tinyurl*
3. Foram excluídos os registros que possuíam metadados que os relacionassem a robôs
4. Foram excluídos os registros provenientes de requisições solicitando algum componente da página (*HEAD* e *favicon*).

Antes da limpeza, os registros possuía cerca de 2 *MB* de dados, após a limpeza foi gerado um arquivo menor com aproximadamente 435,7 *KB* de dados. Por isso pode-se estimar que aproximadamente 78% dos acessos obtidos não eram provenientes de usuários humanos.

5.1.2 Consolidação dos dados

Depois de executar a limpeza dos *logs* de acesso realizados procedimentos para gerar uma tabela com dados para a análise . Primeiramente, construiu-se um *script* em *python* que era capaz de recuperar do *twitter* as informações referentes as contas que receberam um link de acesso ao sítio da pesquisa. Nesta captura constatou-se que algumas contas as quais foram enviados os links foram excluídas.

Foi realizado um cruzamento dos *logs* de acesso e os dados de contas do twitter. Este cruzamento foi realizado de maneira manual utilizando um *software* de planilhas eletrônicas. Após a realização deste cruzamento manual constatou-se que esta não era a melhor maneira de realizar esta operação, pois foi um método muito lento.

Construiu-se um segundo *script* em *python* com o objetivo de obter os dados referentes aos tuítes de pseudo ataques enviados. Esses dados obtidos foram cruzados com os dados do primeiro cruzamento. Mas desta vez o cruzamento foi realizado com auxílio de um *script* em *python* com o auxílio da biblioteca pandas que lida com *dataframes*.

Os dados do cruzamento originaram uma tabela com 7482 linhas, e 19 colunas referentes a dados sobre as contas que receberam os pseudo ataques, o acesso destas ao servidor e qual a mensagem enviada como isca. A tabela apresentava muitos dados repetidos ou até mesmo errados. Para solucionar este problema ocorreu mais uma filtragem manual dos dados utilizando um *software* de planilhas eletrônicas.

Após esta última filtragem obteve-se uma tabela final com 1752 linhas, cada uma com 19 colunas referentes a dados sobre as contas que receberam os pseudo ataques, o acesso destas ao servidor e qual a mensagem enviada como isca. Esta tabela foi responsável pelo embasamento de todas as análises dos dados. As colunas da tabela são as seguintes:

1. ID_CONTA: Número de identificação único da conta dentro da plataforma do twitter
2. ASSUNTO (tema do Bot): Assunto de interesse da conta, identificado pelo uso de uma das palavras-chave
3. name: Nome do usuário da conta na plataforma do twitter
4. lang: Língua que a conta escreve a maioria dos tuítes
5. location: Localização informada pela conta
6. description: Descrição da conta informada pelo usuário
7. followers: número de seguidores da conta
8. friends: número de contas que a conta segue
9. AUTORIDADE: autoridade da conta
10. statuses: número de postagens que a conta fez na plataforma do twitter
11. created: data de criação da conta
12. IP: endereço de IP que realizou o acesso ao sítio da pesquisa
13. screen_name: apelido da conta na plataforma do twitter

14. DT ENVIO ATAQUE: data que foi realizado o envio da isca
15. data acesso : data que realizou o acesso ao sítio da pesquisa
16. DT LEITURA NOTICIA ORIGINAL: data que realizou o acesso à notícia
17. DT ACESSO PROJETO: data que realizou o acesso ao projeto de pesquisa
18. text: mensagem enviada pelo robô para a realização do pseudo ataque
19. bot: apelido da conta que realizou o pseudo ataque

É interessante registrar que interações significativas com o sítio supostamente ocorreram durante o período no qual os *logs* de acesso foram perdidos. Como os dados do banco de dados foram comprometidos estes acessos foram desconsiderados.

Informações foram obtidas através da tabela sobre o acesso ao sítio e cliques no botão "Acessar sem cadastro". Os estímulos enviados geraram a ocorrência de 955 visitas de usuários únicos à página do sítio de *phishing*, e 15 visitas à página de notícias legítimas, após a visita à página do sítio de *phishing*. Foram calculadas as distribuições de frequência das visitas à página de phishing . Para a distribuição de frequência , foram consideradas 30 faixas, a tabela 5.1 mostra as faixas. O gráfico 5.2 foi plotado com as informações da tabela 5.1 utilizando a escala di-log. E é possível ter uma visualização melhor dos dados através dele.

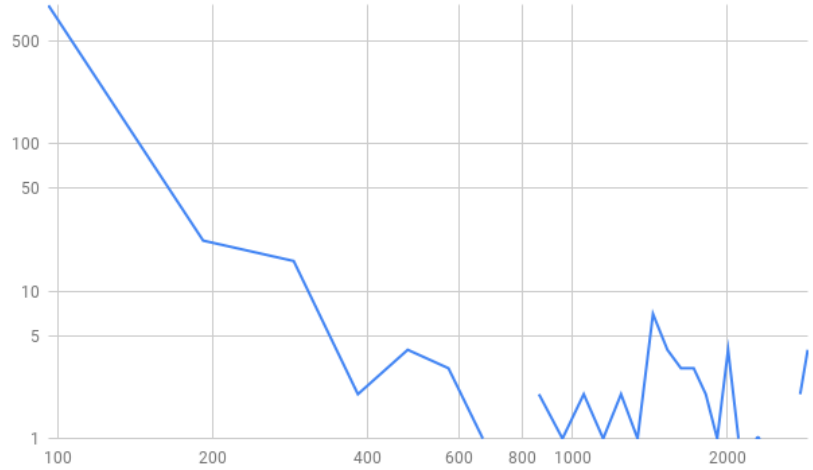

Figura 5.2: Distribuição de frequências do *delay* entre o acesso ao sítio da pesquisa e o momento que o pseudo ataque é enviado nos experimentos 3 e 4, utilizando a escala di-log.

Por sua vez, para a distribuição de frequência do *delay* entre o momento de acesso a notícia (através do botão "Acessar sem cadastro") e o momento que o pseudo ataque é

Limite superior do intervalo em minutos	**Ocorrências**
95,6	866
191,2	22
286,8	16
382,4	2
478,0	4
573,6	3
669,2	1
764,8	0
860,4	2
956,0	1
1051,7	2
1147,3	1
1242,9	2
1338,5	1
1434,1	7
1529,7	4
1625,3	3
1720,9	3
1816,5	2
1912,1	1
2007,7	4
2103,3	1
2198,9	0
2294,5	1
2390,1	0
2485,7	0
2581,3	0
2676,9	0
2772,6	2
2868,2	4
total de respostas aos estímulos	**955**

Tabela 5.1: Distribuição de frequências do *delay* entre o momento que o pseudo ataque é enviado e o acesso ao sítio da pesquisa nos experimentos 3 e 4.

Limite superior do intervalo em minutos	Ocorrências
0,3366666658	6
0,5899999997	1
0,8433333336	3
1,096666667	2
1,350000001	3
total de respostas aos estímulos	**15**

Tabela 5.2: Distribuição de frequências do *delay* entre o acesso a notícia e o momento que o pseudo ataque é enviado nos experimentos 3 e 4.

	coef	Erro Padrão	P>|z|
Interessado no assunto política	1.7296	0.130	0.000
Interessado no assunto esportes	1.5902	0.140	0.000
Número de contas seguidas	3.353E-05	1.92e-05	0.080
Tempo em dias de existência da conta	-0.0003	4.28e-05	0.000
Informou a cidade no perfil	-0.7160	0,119	0.000

Tabela 5.3: Distribuição logística aplicada sobre os dados dos experimentos 3 e 4

enviado foram consideradas 5 faixas. A tabela 5.2 nos mostra como o tempo se distribuiu dentro das faixas.

5.1.3 Modelo para classificação de usuários vulneráveis

Foi executada uma regressão logística sobre a tabela final, onde buscava-se encontrar um modelo que pudesse prever o comportamento dos usuários que acessaram o link enviado. O calculo e a plotagem da regressão logística foi feito utilizando *python* e baseado no *script* presente em [45] utilizando as bibliotecas pandas, numpy e matplotlib.

O resultado da regressão pode ser visto na tabela 5.3, onde a coluna coef corresponde ao coeficiente de correlação do atributo, a coluna erro padrão remete ao erro padrão que aquele atributo possui dentro da regressão e a coluna P>|z| mostra o valor-p. Foi calculada também a curva *ROC* (*receiver operator characteristic*) e a área sob a curva (*AUC*) para avaliar a qualidade da regressão logística. Estes dados encontram-se na figura 5.3.

5.2 Análise dos Resultados

Nesta seção serão analisados os resultados. A Análise é dividida em duas partes. Em uma primeira parte é analisada a evolução do software desenvolvido durante a pesquisa. Em seguida são analisados os dados obtidos através dos experimentos conduzidos.

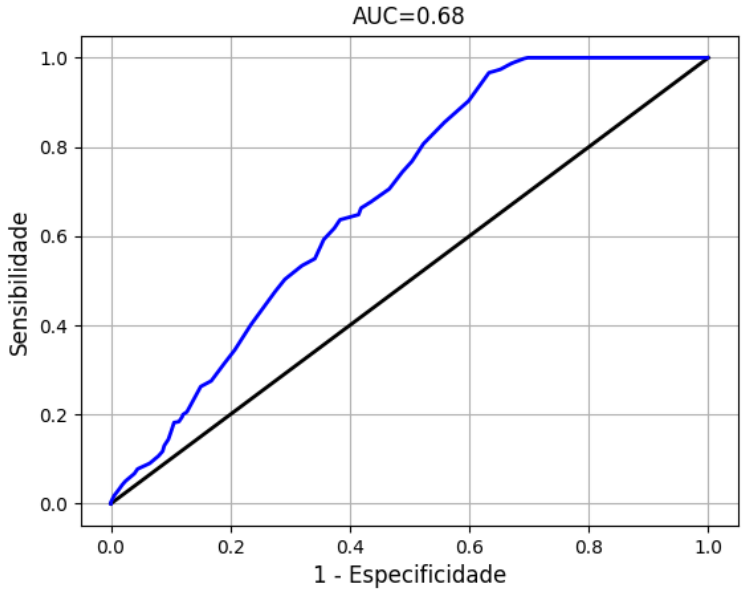

Figura 5.3: Curva *ROC* e a área sob a curva (*AUC*).

5.2.1 Evolução do software

Certamente um dos principais artefatos desenvolvidos durante este trabalho foi o software que realizava os pseudo ataques de *phishing*. A evolução do programa está registrada dentro do capítulo 4. É importante destacar que apesar do experimento 44.4 ter sido bem sucedido o software ainda possui muito a evoluir. Este processo incremental está previsto no trabalho de [7], onde o ciclo de vida de um sistema de engenharia social automatizada conta com um passo de evolução. Neste passo são listados todas as melhorias que puderam ser percebidas durante a execução, e estas são implementadas na próxima versão do sistema.

A evolução do sistema que realiza os pseudo ataques aqui é perceptível ao analisar o software em vários aspectos diferentes. Quando o sistema é visualizado em um mais alto nível de abstração é notável a diferença entre versões, onde a arquitetura evolui de um sistema praticamente monolítico (figura 4.9) para um sistema mais modularizado (figura 4.14) onde cada módulo realiza suas atividades específicas. Esta evolução vislumbrou alcançar requisitos de engenharia de software, como por exemplo a manutenibilidade e o aumento de coesão entre módulos.

Ao olhar o sistema de um ponto de vista mais em um nível de abstração mais baixo, vemos como as funcionalidades mudaram para cumprir requisitos funcionais básicos, tais quais se camuflar perante a plataforma e usuários reais, enviar as iscas. Em grande parte estes requisitos eram referentes as barreiras impostas pelo *twitter* que buscam evitar que

Área Temática	**Desvio padrão**
Entretenimento	18938.36076
Esporte	3914.098048
Política	7896.800311
Total	**7642.562693**

Tabela 5.4: Desvio padrão das distribuições de frequência do experimento 1 (seção 4.1) segmentadas por área temática

robôs maliciosos executem com base na *API*. Entretanto, os robôs desta pesquisa, após todo o aprendizado obtido durante o processo incremental, funcionaram durante 9 dias sem levantar suspeitas da plataforma. Isto é uma forte evidencia de que o twitter ainda pode ser explorado como um vetor para o *phishing*.

Os passos de execução pré-estabelecidos pelos pesquisadores durante a metodologia, se mostraram adequados para este tipo de problema. As mudanças nos passos de execução foram bem sutis. Mas apesar de sutis elas foram muito efetivas na camuflagem dos *BOTs*. Conforme descrito durante a descrição da evolução do *software* no decorrer do capítulo 4

5.2.2 Dados provenientes do *twitter*

O capítulo dos resultados é dividido em seções de acordo com os experimentos performados. Em cada experimento houveram mudanças no software. Isto ocorreu porque o modelo de desenvolvimento adotado foi o modelo incremental. Este processo incremental está previsto no modelo para ataques utilizando engenharia social automatizada criado por [7]. Este modelo encontra-se descrito na seção 2.4. Desta forma cada experimento produziu resultados distintos, que merecem ser analisados separadamente.

O experimento 1 (seção 4.1) coletou dados para caracterizar a distribuição. Estes dados estão presentes nos gráficos das figuras 4.3, 4.4, 4.2 e 4.5. Os gráficos foram plotados para ter uma melhor visualização sobre como se distribuem os usuários no twitter. Para isso, a plotagem foi referente a uma distribuição de frequências das autoridades segmentada em 100 faixas diferentes.

Estes dados são bem heterogêneos como mostra a tabela 5.4, onde estão apresentados os desvios padrão segmentados por áreas de conhecimento. Como os desvios padrão das amostras de usuários são consideravelmente grandes isto nos reflete que os dados coletados são bastante diversificados. Conjuntamente com a ordem de grandeza do volume de dados coletados mostrado na tabela 4.2. É possível assumir que os dados coletados são suficientes para descrever o comportamento dos usuários do twitter que realizaram postagens utilizando alguma das palavras-chave, segmentados com base em suas autoridades.

Faixa de autoridade	Número de ocorrências
< 0.5	371603
>0.5	219125

Tabela 5.5: Distribuição das contas no assunto esportes com autoridades maiores ou menores que 0.5.

As distribuições de frequência dos usuários nas áreas política e entretenimento (figuras 4.2 e 4.4) possuem uma leve semelhança com a curva de distribuição normal, com exceção de 3 picos em autoridades muito baixas muito altas e médias. Suspeita-se que este comportamento dissonante tenha sido causado por contas que realizam postagens automáticas a todo instante, o que é uma conduta diferente do comportamento de usuários humanos. Já os usuários de esportes (figura 4.3) possuem uma distribuição maior em autoridades menores que 0.5 quando comparado com autoridades maiores que 0.5, como nos mostra a tabela 5.5.

Apesar da diferença entre a quantidade de dados coletada (evidenciada na tabela 4.2 que quantifica o volume dos dados e no gráfico 4.5 que compara as curvas de distribuição) os dados coletados são importantes para verificar se não há um viés inserido pela distribuição dos usuários na rede. Por exemplo, determinado grupo ser considerado mais vulnerável apenas por ser maioria, o que por acaso pode vir a ser verdade. Entretanto, o objetivo deste trabalho é de mapear da maneira mais fidedigna possível.

No experimento 2 (seção 4.2) foram realizados os experimentos enviando as iscas para os usuários humanos. Esses ocorreram em apenas algumas horas, devido a um bloqueio automático imposto pelo *twitter* conforme descrito na seção 4.2. Utilizando 2 contas diferentes foram enviados 65 pseudo ataques (32 em uma conta e 33 em outra) para usuários variados, segmentados em 10 faixas de autoridades. Os pesquisadores optaram por não realizar testes com o assunto política, pois esses foram realizados durante o período eleitoral.

Apesar dos dados do experimento 2 serem insignificantes para uma investigação mais profunda e impactante alguns detalhes merecem ser analisados. O gráfico da figura 4.12 sugere que as autoridades maiores tendem a gerar mais interação com a isca enviada. Além disso o comportamento deste gráfico é diferente dos comportamentos dos gráficos das figuras 4.4 e 4.3, o que pode levar a conjecturar que a autoridade afeta positivamente o acesso ao link.

Conforme Já descrito na seção 4.3 os dados dos experimentos 3 e 4 foram analisados de maneira conjunta. A Justificativa para isto é de que não houveram mudanças na metodologia que tornassem esta análise inviável. Esta junção foi benéfica pois é possível ter uma análise de dados mais impactante com uma maior massa de dados. Através da plotagem do gráfico da figura 5.1 é observável que o *delay* entre o acesso ao sítio

da pesquisa e o momento que o pseudo ataque é enviado nos experimentos, retrata um decaimento onde os usuários tendem a clicar na isca logo nos primeiros instantes seguintes ao envio desta. Quanto mais tempo passou desde o envio da isca, menores são as chances do usuário clicar.

De maneira semelhante o *delay* entre o acesso à página e o clique no botão "Acessar sem cadastro"também teve a tendência de decair conforme o tempo passa. Quando plota-se a distribuição de frequências com 3 faixas isto fica bem claro. Uma distribuição em menos classes permite uma visualização melhor do comportamento, devido ao baixo número de interações com a página.

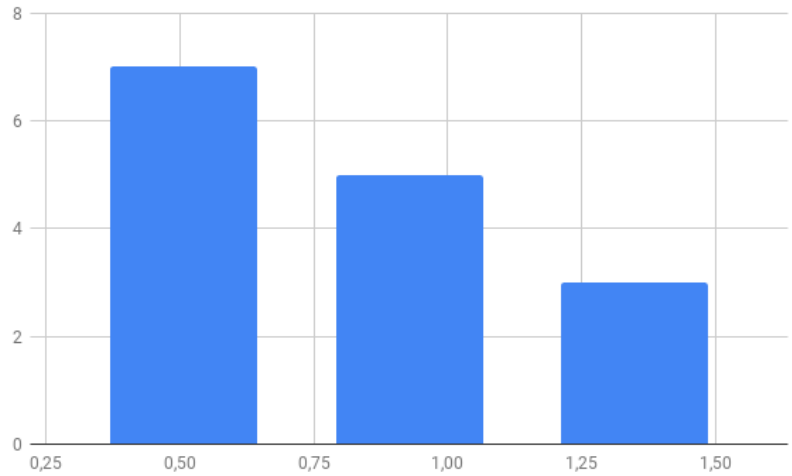

Figura 5.4: Distribuição de frequências do *delay* entre o acesso à notícia e o momento que o pseudo ataque é enviado nos experimentos 3 e 4 em menos classes..

Regressão logística

Objetivando investigar as tendencias que levaram os usuários a acessar o sítio da pesquisa através de uma isca, executou-se uma regressão logística. Resumidamente uma regressão logística é uma técnica estatística que por meio de obervações busca produzir um modelo que tenta prever os valores assumidos por uma variável categórica. Esta técnica é bastante usada em várias áreas de conhecimento é acredita-se que ela é aplicável para este tipo de problema.

Devido ao problema ocorrido que prejudicou a integridade dos dados armazenados no SGBD. Nenhum dado referente ao preenchimento do formulário foi coletado. Entretanto, ainda é possível realizar uma análise sobre a vulnerabilidade dos usuários em clicar no link enviado por um usuário desconhecido. Baseando-se nisto a variável que assumiu o valor categórico aqui foi o acesso do usuário ao sítio da pesquisa.

Foram criados vários modelos de regressão logística, envolvendo os seguintes atributos das contas:

1. Se o usuário que utilizou esta conta realizou alguma postagem envolvendo uma palavra-chave referente a política;
2. Se o usuário que utilizou esta conta realizou alguma postagem envolvendo uma palavra-chave referente a esportes;
3. Se o usuário que utilizou esta conta realizou alguma postagem envolvendo uma palavra-chave referente a entretenimento;
4. O número de seguidores da conta;
5. O número de contas que esta conta segue;
6. O número de postagens que a conta realizou até o momento da coleta;
7. O tempo em dias de existência da conta, desde sua criação até o dia da realização da regressão;
8. O número entre 0 e 1 referente a autoridade da conta;
9. Se a conta informou algo referente a sua localização.

Conforme foi supracitado o calculo e a plotagem da regressão logística foi feito baseado no *script* presente em [45]. Para chegar aos atributos utilizados na tabela 5.3 foram utilizados os seguintes passos:

1. Foi realizado o calculo da regressão logística utilizando todos os atributos das contas que estavam disponíveis;
2. Eliminou-se os atributos que possuíam o valos de $P>|z|$ maior que 0,1;
3. Foi realizado o calculo da regressão logística utilizando permutações dos atributos restantes;
4. Foram descartadas as regressões que não convergiram;
5. Das regressões restantes foi escolhida a que possuia a maior AUC, ou seja, que realizava a melhor previsão.

O modelo que possuiu resultados mais satisfatórios é o apresentado na tabela 5.3. Os dados da tabela mostram que o assunto de interesse da conta possui maior influência na ação de acessar o sítio através da isca. O assunto política tem uma influência maior sobre os usuários do que o assunto esportes. O número de contas seguidas possui uma

influência positiva fraquíssima sobre a ação. E o tempo de existência da conta possui uma influência negativa fraquíssima sobre a ação. Já o fato do usuário informar ou não algo referente a localização possui uma influência negativa e fraca sobre o fenômeno observado, de modo que quando o usuário fornece esta informação menor a chance dele acessar o link informado na isca. A fórmula 5.1 é o resultado da regressão logística(vide seção 2.7), e é capaz de mapear as contas supostamente mais vulneráveis a ataques de *phishing*. Como nenhum dos valores-P é maior que 0.1, a nossa regressão possui um intervalo de confiança de 90%.

$$\frac{1}{1 + e^{-(1.7296x_0 + 1.5902x_1 + 3.353\times10^{-5}x_2 - 0.0003x_3 - 0.7160x_4)}} \tag{5.1}$$

Para confirmar que o modelo descrito acima é o que apresenta resultados mais satisfatórios foi calculada curva *ROC*, que serve para ilustrar o desempenho de um classificador binário. A *AUC* é a área que se encontra abaixo da curva *ROC* e demonstra o quando um classificador binário é melhor que a aleatoriedade. quanto mais próxima a *AUC* está de 1 (onde 1 seria um classificador perfeito) melhor é o classificador. Como mostrado na figura 5.3 a *AUC* do classificador gerado neste trabalho é igual a 0.68. Portanto apesar do classificador precisar de algum refino, ele é melhor na identificação da vulnerabilidade do que escolher um alvo ao acaso.

5.3 Discussão dos Resultados

Dentre os trabalhos de outros pesquisadores estudados para a concepção desta pesquisa, o artigo científico encontrado que mais se assemelha a proposta desta pesquisa é o trabalho de [13]. Em comparação com o artigo [13], o trabalho descrito neste documento supostamente apresentou um maior número de acessos à página da pesquisa em menos tempo. Este fato não pode ser afirmado com maior veemência pois os pesquisadores [13] fizeram uma filtragem mais profunda nos *logs* de acesso do sítio deles. Esta filtragem ocorreu verificando cada endereço de *IP* que contava no *log* através do sistema *WHOIS*. Isto permitiu que eles agrupassem *IPs* provenientes de um mesmo domínio, o que pode ser um indicativo de um acesso de um usuário não humano.

Entretanto, caso este experimento tenha de fato alcançado um maior número de acessos, este fato pode vir a ser justificado devido a estratégia de envio de iscas, aqui a estratégia foi um pouco mais agressiva. As iscas foram enviadas mencionando os alvos, o que pode causar mais curiosidade nas vítimas dos pseudo ataques. A interação dos robôs com os usuários também foi observada durante ao experimento de modo análogo ao mostrado em [13]. Entretanto aqui não foi coletada a informação acerca da quantidade de seguidores, ou interações que eram provenientes de outros robôs.

O fenômeno da indiscrição não pôde ser observado durante o experimento, pois não haviam *logs* de acesso referentes ao preenchimento do formulário durante o período ao qual os *logs* puderam ser recuperados. Apesar disto é importante discutir sobre a vulnerabilidade que estão expostos os usuários que acessaram o link informado por um terceiro totalmente desconhecido.

Quanto aos *delays* observados entre o envio da isca e o acesso a página, e entre o acesso a página e o clique no botão "acessar sem cadastro". Em ambos houve uma tendência de decaimento. No caso do *delay* da entre o envio da isca e o acesso a página era um comportamento esperado por se tratar de manchetes de notícias, onde naturalmente as notícias ficam ultrapassadas com o tempo. Já o *delay* entre acesso a página e o clique no botão "acessar sem cadastro", pode ser explicado pela falta de elementos no sítio da pesquisa que pudessem servir de distração, onde a decisão de qual ação tomar seria muito rápida. Houve ainda um acesso ao projeto da pesquisa, entretanto ele não serviu para maiores análises.

É necessário entender que são níveis completamente diferentes de uma suposta vulnerabilidade. Pode se assumir que quem somente clica em um link enviado por desconhecidos é menos vulnerável do que quem interage com um site o qual não se sabe ser seguro, por sua vez quem fornece dados neste mesmo site é o mais vulnerável deste grupo.

Por isso, para mensurar qual o grupo de usuários supostamente era o mais vulnerável optou-se pelo cálculo de uma regressão logística. Este tipo de regressão é aplicado na literatura estudada, como por exemplo em [10]. Foram criados diversos modelos antes de chegar no modelo descrito pela tabela 5.3.

Os resultados encontrados através da regressão logística evidenciam o que a teoria sobre engenharia social aborda. Segundo [9] os engenheiros sociais obtém êxito através de 3 mecanismos assimetria epistêmica, dominância tecnocrática e substituição teleológica. Neste trabalho assimetria epistêmica foi evidenciada pela influencia negativa do tempo de existência da conta no twitter na resposta preliminar ao ataque. Onde quanto mais o usuário conhece o twitter menor a chance dele cair no golpe. A dominância tecnocrática dos pesquisadores foi demonstrada por meio do uso de robôs dentro da plataforma mapeando usuários vulneráveis aos pseudo ataques. Destarte através dos mecanismos já discutidos conseguiu-se fazer a substituição teleológica da plataforma, através dos pseudo ataques.

Acredita-se que devido ao menor número de ocorrências, as distribuições realizadas com o atributo referente a realização de postagens com palavras-chave referentes a entretenimento não convergiram. O uso da autoridade não resultou em um modelo bom. Quando este atributo foi utilizado os modelos de regressão convergiram, mas apresentaram o valor de AUC muito próximo de 0.5 o que invalida o uso da regressão com a autoridade.

Por outro lado a informação da localização pelo o usuário da conta surpreendentemente apresentou um impacto significativo nas chances do usuário clicar ou não no link.

Conforme já foi citado na seção 2.4, uma das principais contribuições de [7] foi a concepção de um ciclo de execução para sistemas de engenharia social automatizada. Este trabalho encaixa-se perfeitamente neste ciclo. Cada experimento esteve dividido nas 5 fases propostas em [7] Planejamento, Mapeamento de alvos, Execução, Recrutamento, Evolução:

- Experimento 1: Concepção do primeiro modelo dos robôs,Execução da rotina de busca por autoridade,Envio de iscas baseando-se nas faixas de autoridade,Divulgação da isca pela vítima (através de curtida ou retuíte),análise dos erros e acertos
- Experimento 2: implementação das melhorias, com base no aprendizado do experimento anterior",Execução da rotina de busca por autoridade,Envio de iscas baseando-se nas faixas de autoridade,Divulgação da isca pela vítima (através de curtida ou retuíte),análise dos erros e acertos
- Experimento 3: implementação das melhorias, com base no aprendizado do experimento anterior",Execução da rotina de busca por autoridade,Envio de iscas baseando-se nas faixas de autoridade,Divulgação da isca pela vítima (através de curtida ou retuíte),análise dos erros e acertos
- Experimento 4: implementação das melhorias, com base no aprendizado do experimento anterior",Execução da rotina de busca por autoridade,Envio de iscas baseando-se nas faixas de autoridade,Divulgação da isca pela vítima (através de curtida ou retuíte),análise dos erros e acertos

Capítulo 6

Conclusão

Neste capítulo 6 serão explanados quais dos objetivos traçados foram alcançados, quais os problemas encontrados e as soluções propostas para estes. Além disto no final do capítulo são traçadas as perspectivas de trabalhos futuros.

6.1 Objetivos Propostos e Alcançados

Esta pesquisa tinha como objetivo geral desenvolver um modelo matemático-computacional para explicar o fenômeno da vulnerabilidade do usuário de uma conta de ser indiscreto no fornecimento de dados sensíveis na sua relação com estranhos, na mídia social *Twitter*. Para fornecer respostas às questões, o trabalho conduziu pseudo-ataques de *phishing*, junto a contas de usuários do Twitter, cuja autoridade pôde ser calculada previamente, a fim de verificar se há uma correlação entre a autoridade dessa conta e a propensão de seu usuário em fornecer informações supostamente sensíveis a um desconhecido (o pseudo-atacante).

O objetivo geral foi parcialmente alcançado. Conseguiu-se obter um modelo matemático-computacional que conseguia prever com uma precisão melhor do que escolher alvos ao acaso quais seriam os usuários supostamente mais vulneráveis com base em suas informações obtidas no *twitter*. Entretanto, conforme já foi descrito mais detalhadamente nos capítulos 4 e 5 o fenômeno da indiscrição não foi capturado durante a execução dos experimentos. Talvez isso tenha ocorrido devido ao fato da isca não ser atrativa o suficiente para este tipo de ataque de *phishing*. Uma hipótese para essa baixa atratividade da isca pode ser o fato de se tratar de uma notícia. O fato de se tratar de uma notícia pode tornar mais atrativo para as vítimas do pseudo ataque procurar outra fonte de informação do que preencher as informações.

Por isso, a suposta vulnerabilidade observada no trabalho foi referente somente ao ato de clicar em um link enviado por um usuário desconhecido. Para possibilitar a observação

do objetivo geral, o trabalho foi subdividido em 2 (dois) objetivos específicos, e mediante o cumprimento destes, vislumbrou-se alcançar o objetivo principal da pesquisa:

1. Investigar quantitativamente (através de métodos estatísticos) se existe uma correlação entre a autoridade de uma conta no twitter e a propensão do usuário desta conta a fornecer informações supostamente sensíveis a desconhecidos;
2. Verificar se, em grupos de contas com assuntos de interesse diferentes (relacionadas a política, religião ou entretenimento) a correlação sondada no item 1 pode ser observada de maneira significativa;

A partir dos objetivo específicos, podemos obter algumas conclusões. O primeiro objetivo específico não foi alcançado. Não houveram evidências empíricas que fossem suficientemete fortes para ratificar que existe alguma correlação entre a autoridade de uma conta no twitter e a vulnerabilidade desta a ataques de *phishing*.O segundo objetivo foi parcialmente cumprido. Afirma-se isso porque durante o cálculo da regressão logística observou-se que o assunto de interesse possui uma leve correlação com o a suposta vulnerabilidade do usuário.

6.2 Problemas encontrados e soluções propostas

Dentro do capítulo 4 são discutidos os diversos problemas que ocorreram durante os experimentos juntamente com suas soluções. Apesar disso no experimento final não há soluções propostas devido a interrupção do processo incremental de desenvolvimento do *software*. Esses foram a ausência do armazenamento de alguns outros dados durante o processo de experimentação e a a falha estratégica em persistir os acessos utilizando um sistema gerenciador de bancos de dados.

Solucionar esses problemas em uma nova iteração do desenvolvimento do *software* seria de certa forma trivial. Invés de utilizar um SGBD, bastaria realizar uma gerência dos *logs* de acesso disponibilizados pelo servidor. Quanto aos dados, em uma nova iteração do desenvolvimento serão guardadas todas as informações disponíveis.

Uma melhoria nas iscas poderia vir a tornar os pseudo-ataques mais expressivos. Aparentemente as notícias não foram atrativas o suficiente para que os alvos dos pseudo-ataques se sentissem razoavelmente tentados a fornecer informações supostamente sensíveis ao sítio da pesquisa.

6.3 Perspectivas de trabalhos futuros

Apesar dos objetivos traçados não terem sido alcançados completamente, este trabalho contribuiu para um avanço na fronteira do conhecimento acerca de sistemas de engenharia social automatizada. Os acertos podem vir a ser utilizados em futuros experimentos relacionados ao tema. E os erros podem ser utilizados como um conjunto de más práticas, de modo que experimentos futuros possam evitar os erros aqui cometidos.

Fica como perspectiva de trabalhos futuros repetir o experimento durante um período maior de tempo. Outros aspectos podem ser inseridos em uma nova versão do experimento tais como executá-lo em outras plataformas, observando outros atributos dos usuários, cruzando informações entre as plataformas, usar algoritmos de inteligência artificial mais robustos (tornando os robôs mais humanos). Talvez com essas melhorias seja possível mapear aspectos que indiquem a suposta vulnerabilidade de modo mais contundente.

Referências

[1] TANENBAUM, ANDREW S e Sistemas Operacionais Modernos: *edição*, 3. xii, 2, 8, 9, 14

[2] Técnicas, Associação Brasileira De Normas: *ABNT NBR ISO/IEC 17799: 2005: tecnologia da informação-técnicas de segurança-código de prática para a agestão da segurança da informação*. ABNT, 2005. 2

[3] Mitnick, Kevin D e William L Simon: *A arte de enganar*. São Paulo. Person Education do Brasil Ltda, 2003. 2, 3, 9, 10

[4] Schneier, Bruce: *Secrets & lies: Digital security in a networked world*. International Hydrographic Review, 2(1):103–104, 2001. 2

[5] Arce, Ivan: *The weakest link revisited [information security]*. IEEE Security & Privacy, 99(2):72–76, 2003. 2

[6] McGregor, Susan E, Elizabeth Anne Watkins, Mahdi Nasrullah Al-Ameen, Kelly Caine e Franziska Roesner: *When the weakest link is strong: Secure collaboration in the case of the panama papers*. Em *th USENIX Security Symposium (USENIX Security)*, 2017. 2

[7] Huber, Markus, Stewart Kowalski, Marcus Nohlberg e Simon Tjoa: *Towards automating social engineering using social networking sites*. Em *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 3, páginas 117–124. IEEE, 2009. 2, 3, 11, 12, 62, 63, 69

[8] Thornburgh, Tim: *Social engineering: the dark art*. Em *Proceedings of the 1st annual conference on Information security curriculum development*, páginas 133–135. ACM, 2004. 2

[9] Hatfield, Joseph M: *Social engineering in cybersecurity: The evolution of a concept*. Computers & Security, 73:102–113, 2018. 3, 6, 9, 10, 68

[10] Oliveira, Daniela, Harold Rocha, Huizi Yang, Donovan Ellis, Sandeep Dommaraju, Melis Muradoglu, Devon Weir, Adam Soliman, Tian Lin e Natalie Ebner: *Dissecting spear phishing emails for older vs young adults: On the interplay of weapons of influence and life domains in predicting susceptibility to phishing*. Em *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, páginas 6412–6424. ACM, 2017. 3, 11, 21, 68

[11] Jakobsson, Markus e Jacob Ratkiewicz: *Designing ethical phishing experiments: a study of (rot13) ronl query features*. Em *Proceedings of the 15th international conference on World Wide Web*, páginas 513–522. ACM, 2006. 3, 11

[12] Jakobsson, Markus, Nathaniel Johnson e Peter Finn: *Why and how to perform fraud experiments*. IEEE Security & Privacy, 6(2), 2008. 3, 5, 6, 11, 22, 34

[13] Shafahi, Mohammad, Leon Kempers e Hamideh Afsarmanesh: *Phishing through social bots on twitter*. Em *Big Data (Big Data), 2016 IEEE International Conference on*, páginas 3703–3712. IEEE, 2016. 3, 4, 6, 11, 67

[14] Lauinger, Tobias, Veikko Pankakoski, Davide Balzarotti e Engin Kirda: *Honeybot, your man in the middle for automated social engineering*. Em *LEET*, 2010. 3, 11

[15] Purkait, Swapan: *Phishing counter measures and their effectiveness–literature review*. Information Management & Computer Security, 20(5):382–420, 2012. 3, 5, 6, 7, 11

[16] Lastdrager, Elmer EH: *Achieving a consensual definition of phishing based on a systematic review of the literature*. Crime Science, 3(1):9, 2014. 3

[17] Mell, Peter: *Nist special publication 800-83. guide to malware incident prevention and handling*, 2005. 3

[18] Dhamija, Rachna, J Doug Tygar e Marti Hearst: *Why phishing works*. Em *Proceedings of the SIGCHI conference on Human Factors in computing systems*, páginas 581–590. ACM, 2006. 3

[19] Oliveira, Filipe: *Brasil tem o 3o maior crescimento do twitter em número de usuários*. `http://www1.folha.uol.com.br/tec/2017/02/1861175-numero-de-usuarios-do-twitter-no-brasil-cresce-18-em-2016.shtml`, 2017. Recuperado 2 de junho de 2018. 4

[20] Nagmoti, Rinkesh, Ankur Teredesai e Martine De Cock: *Ranking approaches for microblog search*. Em *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, páginas 153–157. IEEE, 2010. 4, 20, 21

[21] Alencar, Gliner Dias, MF Lima e André CA Firmo: *O efeito da conscientização de usuários no meio corporativo no combate à engenharia social e phishing*. IX Simpósio Brasileiro de Sistemas de Informação (SBSI'13), páginas 254–259, 2013. 5, 6

[22] *7 signs we are too dependent on technology*. `https://www.mnn.com/green-tech/gadgets-electronics/stories/7-signs-we-are-too-dependent-on-technology`, acesso em 2018-12-17. 8

[23] *Celular já substitui documentos como carteira de trabalho, CNH, CPF e título de eleitor*. `https://extra.globo.com/noticias/economia/celular-ja-substitui-documentos-como-carteira-de-trabalho-cnh-cpf\penalty\z@-titulo-de-eleitor-22152110.html`, acesso em 2018-12-17. 8

[24] Patel, Dipti e Xin Luo: *Take a close look at phishing*. Em *Proceedings of the 4th annual conference on Information security curriculum development*, página 32. ACM, 2007. 11

[25] Cisco Systems, Inc.: *Cisco 2017 annual cybersecurity report*. `https://www.cisco.com/c/dam/m/digital/1198689/Cisco_2017_ACR_PDF.pdf`, 2017. 11

[26] Souza, Raul Carvalho de e Jorge Henrique Cabral Fernandes: *Um estudo sobre a confiança em segurança da informação focado na prevenção a ataques de engenharia social nas comunicações digitais*. Brazilian Journal of Information Science, 10(1), 2016. 11

[27] *The Importance of #Hashtags*. `https://www.socialmediatoday.com/content/importance-hashtags`, acesso em 2018-12-17. 13

[28] Fielding, Roy T e Richard N Taylor: *Architectural styles and the design of network-based software architectures*, volume 7. University of California, Irvine Irvine, USA, 2000. 14

[29] Group, W3C Working *et al.*: *Web services architecture*. http://www. w3. org/TR/ws-arch/, 2004. 14

[30] *API reference index*. `https://developer.twitter.com/en/docs/api-reference-index.html`, acesso em 2018-12-17. 14

[31] *Developer terms*. `https://developer.twitter.com/en/developer-terms.html`, acesso em 2018-11-21. 15, 25

[32] Regulation, General Data Protection: *Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46*. Official Journal of the European Union (OJ), 59(1-88):294, 2016. 15

[33] *API Reference - tweepy 3.5.0 documentation*. `http://docs.tweepy.org/en/v3.5.0/api.html#API.search`, acesso em 2018-12-17. 16, 17

[34] *Streaming With Tweepy - tweepy 3.5.0 documentation*. `http://docs.tweepy.org/en/v3.5.0/streaming_how_to.html`, acesso em 2018-12-17. 17

[35] *Tweet object*. `https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object.html`, acesso em 2018-12-01. 18

[36] *User object*. `https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/user-object.html`, acesso em 2018-12-01. 19

[37] Swaminathan, Saishruthi: *Logistic Regression a Detailed Overview*, março 2018. `https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc`, acesso em 2019-01-16. 21

[38] *Logistic Regression*. http://userwww.sfsu.edu/efc/classes/biol710/logistic/logisticreg.htm, acesso em 2019-01-16. 21

[39] http://dgp.cnpq.br/dgp/espelhogrupo/2334079961242970, acesso em 2018-12-18. 24

[40] *Rock in Rio e 'BBB' são os assuntos mais comentados no Twitter*, dezembro 2017. https://f5.folha.uol.com.br/voceviu/2017/12/rock-in-rio-e-bbb-sao-os-assuntos-mais-comentados-no-twitter.shtml, acesso em 2018-06-27. 28

[41] *Twitter revela principais assuntos e tweets na rede social no Brasil e no mundo em 2017*, dezembro 2017. https://www.b9.com.br/83133/twitter-revela-principais-assuntos-e-tweets-na-rede-social-no-brasil\penalty\z@-e-no-mundo-em-2017/, acesso em 2018-06-27. 28

[42] Solano, E., L. Gê e G. Maringoni: *O ódio como política: a reinvenção das direitas no Brasil*. Coleção Tinta Vermelha. Boitempo Editorial, 2018, ISBN 9788575596555. https://books.google.com.br/books?id=k3lvDwAAQBAJ. 48

[43] Banker, P.: *Neymar*. Planeta, 2014, ISBN 9788542203479. https://books.google.com.br/books?id=Qg6GAwAAQBAJ. 48

[44] Orsi, C.: *O Livro Da Astrologia*. CreateSpace Independent Publishing Platform, 2016, ISBN 9781535313032. https://books.google.com.br/books?id=LDL_vQAACAAJ. 48

[45] Le, Postado por e ro Guerra: *Aplicação de uma regressão logística em Python*. http://artedosdados.blogspot.com/2013/10/aplicacao-de-uma-regressao-logistica-em.html, acesso em 2018-12-21. 61, 66